INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

Year: 2006 Volume: I

Supplementary issue - Proceedings of ICCCC 2006

CCC Publications

www.journal.univagora.ro

Publisher & Editorial Office

CCC Publications

Agora University

Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526

Tel: +40 259 427 398, Fax: +40 259 434 925, E-mail: ccc@univagora.ro

Website: www.journal.univagora.ro

ISSN 1841-9836 (print version)

ISSN 1841-9844 (online version)

AGORA UNIVERSITY

IEEE Computer Society

Editors
Ioan Dziţac Florin-Gheorghe Filip Mişu-Jan Manolescu

PROCEEDINGS
of the
International Conference on Computers, Communications & Control

June 1-3, Băile Felix - Oradea, Romania

ICCCC 2006
www.iccc.univagora.ro

Editors of the Proceedings
Ioan DZITAC, Agora University, Oradea, Head of the Business Informatics Department
Florin-Gheorghe FILIP, Romanian Academy, Vice-President of the Romanian Academy
Mişu-Jan MANOLESCU, Agora University, Oradea, Rector of the Agora University

Managing Editor of the Proceedings
Adriana MANOLESCU, Agora University, Dean of the Law and Economics Faculty

Technical Editor of the Proceedings
Horea OROS

Cover design
Marius CHERECHEŞ

Printed
Metropolis SRL, Oradea, Romania, Tel. +4 0259 472640

The printing of the proceedings was sponsored by the Ministry of Education and Research, Romania-National Authority for Scientific Research.

Editorial Address
CCC Publications
Agora University
St. Piaţa Tineretului No. 8
Oradea, jud. Bihor, Romania
Zip Code 410526
Tel: +40 259 427 398
Fax: +40 259 434 925
E-mail: ccc@univagora.ro
Website: www.journal.univagora.ro

CCC Publications, powered by Agora University Publishing House, currently publishes the "International Journal of Computers, Communications & Control" and its scope is to publish scientific literature (journals, books, monographs and conference proceedings) in the field of Computers, Communications and Control.

INTERNATIONAL PROGRAM COMMITTEE

Grigore ALBEANU, University of Bucharest, ROMANIA
Ilie BABAITA, The West University of Timisoara, ROMANIA
Barnabas BEDE, Budapest Tech, HUNGARY
Vasile BERINDE, North University of Baia Mare, ROMANIA
Alexandru BICA, University of Oradea, ROMANIA
Florian BOIAN, UBB Cluj Napoca, ROMANIA
Valentin CASAVELA, Agora University, ROMANIA
Mitică CRAUS, Technical University of Iasi, ROMANIA
Paul CRISTEA, Politehnica University of Bucharest, ROMANIA
Doina DANAIATA, The West University of Timisoara, ROMANIA
Ioan DESPI, University of New England, AUSTRALIA
Antonio DI NOLA, University of Salerno, ITALY
Dan DUMITRESCU, UBB Cluj Napoca, ROMANIA
Ioan DZIŢAC, AGORA University, Oradea, ROMANIA (Chair)
Ömer EGECIOGLU, University of California, USA
Florin FILIP, Romanian Academy, ROMANIA (Honorary Chair)
Janos FODOR, Budapest Tech, HUNGARY
Militon FRENŢIU, UBB Cluj-Napoca, ROMANIA
Angel GARRIDO, Facultad de Ciencias,UNED,SPAIN
Adelina GEORGESCU, University of Pitesti, ROMANIA
George GEORGESCU, University of Bucharest, ROMANIA
Mircea GIURGIU, Tech. University of Cluj-Napoca, ROMANIA
Dan GRIGORAŞ, University College Cork, IRELAND
Kaoru HIROTA, Tokyo Institute of Technology, JAPAN
Afrodita IORGULESCU, ASE Bucharest, ROMANIA
Toader JUCAN, AIC University of Iasi, ROMANIA
Michail KALOGIANNAKIS, TEI of Crete, GREECE
Mişu-Jan MANOLESCU, AGORA University, Oradea, ROMANIA
Solomon MARCUS, IMAR, Romanian Academy, ROMANIA
Ioan MANG, University of Oradea, ROMANIA
Mircea MARIN, University of Tsukuba , JAPAN
Grigor MOLDOVAN, UBB Cluj Napoca, ROMANIA
Hajime NOBUHARA, Tokyo Institute of Technology, JAPAN
Mohamed NOUR, The Electronics Research Institute of Cairo, EGYPT
Gheorghe PĂUN, IMAR, Romanian Academy, ROMANIA
Mario de J. PEREZ-JIMENEZ, University of Seville, SPAIN
Willi PETERSEN, Universität Flensburg, GERMANY
Bazil PÂRV, UBB Cluj Napoca, ROMANIA
Eugen PETAC, Ovidius University, Constanta, ROMANIA (Vice-Chair)
Dana PETCU, Western University of Timisoara, ROMANIA
Bogdana POP, Transilvania University of Brasov, ROMANIA
Constantin POPESCU, University of Oradea, ROMANIA
Daniela Elena POPESCU, University of Oradea, ROMANIA
Alvaro ROMERO JIMENEZ, University of Seville, SPAIN
Ioan ROXIN, University of Franche-Comte, FRANCE
Imre J. RUDAS, Budapest Tech, HUNGARY
Daniel STAMATE, Goldsmiths University of London, UK
Pantelimon STANICA, Auburn University Montgomery, USA
Athanassios D. STYLIADIS, Alexander Institute of Technology, GREECE
Sabin TABIRCA, University College Cork, IRELAND

Gheorghe TECUCI, George Mason University, USA
Horia-Nicolai TEODORESCU, Technical University of Iasi, ROMANIA
Ioan TOMESCU, University of Bucharest, ROMANIA
Dan TUFIS, RACAI, Romanian Academy, ROMANIA
Michael Gr VASSILAKOPOULOS, TEI of Thessaloniki, GREECE
Gabriel VLADUT, IRC 4D, IPA CIFATT S.A. Craiova, ROMANIA
Doina ZMARANDA, University of Oradea, ROMANIA

ORGANIZING COMMITTEE
Barnabas BEDE, Budapest Tech, HUNGARY
Viorina BERDE, Agora University, ROMANIA
Daniel BRINZAŞ, Jumpeye Creative Media, ROMANIA, (Website designer)
Delia CIURBA, Agora University, ROMANIA
Romulus COSTINAS, Agora University, ROMANIA
Ioan DZIŢAC, Agora University, ROMANIA, (Founder, General Organizer)
Renata DZIŢAC, Agora University, ROMANIA
Simona DZIŢAC, University of Oradea, ROMANIA
Petru FILIP, Agora University, ROMANIA
Mihail FLOROVICI, Romanian Consulate of SERBIA & MONTENEGRO
Marcel GĂITĂNARU, Agora University, ROMANIA
Loredana GALEA, Agora University, ROMANIA
Ramona GANEA, Agora University, ROMANIA
Leon GHEMEŞ, Agora University, ROMANIA, (Webmaster)
Nicoleta MAGHIARI, Agora University, ROMANIA
Adriana MANOLESCU, Agora University, ROMANIA, (Vice-Chair)
Mişu-Jan MANOLESCU, Agora University, ROMANIA, (Chair)
Horea OROS, University of Oradea, ROMANIA, (Secretary of the ICCCC 2006)
Daniela PANTEA, Agora University, ROMANIA
Dorina PETAC, ICT Constanta, ROMANIA
Eugen PETAC, ICT Constanta, ROMANIA
Răducu PURCEL, Agora University, ROMANIA
Athanassios STYLIADIS, ATEI, GREECE
Sabin TABIRCA, University College Cork, IRELAND

PREFACE

The second edition of the International Conference on Computers, Communications & Control[1], ICCCC 2006, was organized by Agora University of Oradea and was powered by IEEE Computer Society, Romania Section, and took place in Baile Felix - Oradea, Romania, June 1-3, 2006.

ICCCC 2006 provides a forum for scientist in academia and industry to present and discuss their latest research findings on a broad array of topics in Computer Science, Information Technology & Data Communications and Computer-based Control.

The scope of the conference covered the following topics: Artificial Intelligence, Automata and Formal Languages, Computational Mathematics, Cryptography and Security, E-Activities, Fuzzy Systems, Informatics in Control, Information Society - Knowledge Society, Natural Computing, Network Design & Internet Services, Multimedia & Communications, Parallel and Distributed Computing.

ICCCC 2006 and the the International Journal of Computers, Communications & Control (IJCCC, founded by I. Dziţac - Executive Editor, F.G. Filip - Editor in Chief and M.J. Manolescu - Managing Editor), celebrates, by two invited papers[2] published in IJCCC Vol. I, No. 1, 100 years from the birth of Grigore C. Moisil (1906-1973). Grigore C. Moisil was one of the great Romanian mathematicians who had a great impact in Computer Science. He received post-mortem, in 1996, the "Computer Pioneer Award" of IEEE Computer Society. He insisted and helped in the building of the first Romanian computer, by Victor Toma, at the Institute of Atomic Physics (1957). He also directed the first generation of graduate students in Mathematics to work with the team of Victor Toma, at the Institute of Atomic Physics; they were trained to learn programming at the new computers CIFA. He introduced Łukasiewicz algebras with three values and multiple values (which are known today as Łukasiewicz-Moisil algebras) and used them in the logic and study of commutation circuits. He developed new methods of analysis for finite automata and had valuable contributions in the filed of algebraic theory of automated mechanism.

The Program Committee received 142 submissions, originating from Algeria, France, Germany, Greece, Hungary, Italy, Japan, India, Ireland, Iran, Spain, Serbia & Montenegro, Moldova, Romania, Thailand, Tunisia and and USA. Each submission was reviewed by two Program Committee members, or other experts. Out of the 142 papers only 91 (64%) were accepted for presentation at the conference and for publication (7 papers in IJCCC, Vol. I (2006), No.1 and 84 papers in this supplementary issue of IJCCC).

The Program Committee gratefully acknowledges all authors who submitted papers for theirs efforts in maintaining the scientific standards of the second edition of ICCCC .

We would like to thank the members of the Program Committee, the additional reviewers and the members of the Organizing Committee for their work and support.

Also, we thank the authors that responded to our request for preparing invited papers: K. Chen, G. Ciobanu, P. D. Cristea, F. Dong, D. Dumitrescu, J. Fodor, A. Garrido, K. Hirota, I.D. Karamitsos, A. Roth, I. Rudas, M. Stanojevici, A.D. Stylidias, and D. Tufiş. M. Vujosevici and D. I. Zacharoiu.

We would like to express our gratitude for their support to:

- Agora University, Oradea, Romania;
- ICT Foundation, Constanta, Romania, especially to president E. Petac;
- IEEE Computer Society, Romania Section, especially to president N. Ţăpuş;
- Ministry of Education and Research, Romania-National Authority for Scientific Research;
- Romanian Academy.

Oradea, May 2006
I. Dziţac, F.G. Filip, M. J. Manolescu

[1] The first edition of this conference, entitled "International Conference on Computers and Communications", ICCC 2004, has been founded and organized in 2004 by I. Dziţac, C. Popescu and H. Oros.

[2] "Grigore C. Moisil (1906 - 1973) and his School in Algebraic Logic", authors George Georgescu, Afrodita Iorgulescu, Sergiu Rudeanu and "Grigore C. Moisil: A Life Becoming A Myth", author Solomon Marcus

Table of contents

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 13-22

A Programming Perspective of the Membrane Systems

Invited paper

Gabriel Ciobanu

Abstract: We present an operational semantics of the membrane systems, using an appropriate notion of configurations and sets of inference rules corresponding to the three stages of an evolution step in membrane systems: maximal parallel rewriting step, parallel communication of objects through membranes, and parallel membrane dissolving.
We define various arithmetical operations over multisets in the framework of membrane systems, indicating their complexity and presenting the membrane systems which implement the arithmetic operations.
Finally we discuss and compare various sequential and parallel software simulators of the membrane systems, emphasizing their specific features.

Keywords: membrane systems, operational semantics, arithmetical operations over multisets.

1 Membrane Systems

Membrane systems represent a computational model inspired by cell compartments and molecular membranes. Essentially, such a system is composed of various compartments, each compartment with a different task, and all of them working simultaneously to accomplish a more general task of the whole system. A detailed description of the membrane systems (also called P systems) can be found in [16]. A *membrane system* consists of a hierarchy of membranes that do not intersect, with a distinguishable membrane, called the *skin membrane*, surrounding them all. The membranes produce a delimitation between *regions*. For each membrane there is a unique associated region. Regions contain multisets of *objects*, *evolution rules* and possibly other membranes. Only rules in a region delimited by a membrane act on the objects in that region. The multiset of objects from a region corresponds to the "chemicals swimming in the solution in the cell compartment", while the rules correspond to the "chemical reactions possible in the same compartment". Graphically, a membrane structure is represented by a Venn diagram in which two sets can be either disjoint, or one is a subset of the other. More details (concepts, results) and several variants of membrane systems are presented in [16].

A *P system* consists of several membranes that do not intersect, and a *skin membrane*, surrounding them all. The membranes delimit *regions*, and contain multisets of *objects*, as well as *evolution rules*. Each membrane has a unique associated region. The space outside the skin membrane is called the *outer region* (or the environment). Because of the one-to-one correspondence between the membranes and the regions, we usually use the word membrane instead of region. Only rules in a region delimited by a membrane act on the objects in that region. Moreover, the rules must contain target indications, specifying the membrane where objects are sent after applying the rule. The objects can either remain in the same region, or pass through membranes in two directions: they can be sent *out* of the membrane which delimits a region from outside, or can be sent *in* one of the membranes which delimit a region from inside, precisely identified by its label. The membranes can also be *dissolved*. When such an action takes place, all the objects of the dissolved membrane remain free in the membrane placed immediately outside, but the evolution rules of the dissolved membranes are lost. The skin membrane is never dissolved. The application of evolution rules is done in parallel, and it is eventually regulated by *priority* relationships between rules. A *P system* of degree m is a structure $\Pi = (O, \mu, w_1, \ldots, w_m, (R_1, \rho_1), \ldots, (R_m, \rho_m), i_o)$, where:

(i) O is an alphabet of objects, and μ is a membrane structure;

(ii) w_i are the initial multisets over O associated with the regions defined by μ;

(iii) R_i are finite sets of evolution rules over O associated with the membranes, of typical form $u \rightarrow v$, with u a multiset over O and v a multiset containing paired symbols (messages) of the form $(c, here)$, (c, in_j), (c, out) and the dissolving symbol δ;

(iv) ρ_i is a partial order relation over R_i, specifying a *priority* relation among the rules: $(r_1, r_2) \in \rho_i$ iff $r_1 > r_2$ (i.e., r_1 has a higher priority than r_2);

(v) i_0 is either a number between 1 and m specifying the *output* membrane of Π, or it is equal to 0 indicating that the output is the outer region.

Since the skin is not allowed to be dissolved, we consider that the rules of the skin do not involve δ. These are the *general P systems*, or *transition P systems*; many other variants and classes were introduced [16].

The existing results regarding the P systems refer mainly to their computation power and complexity, namely to their characterization of Turing computability (universality is obtained even with a small number of membranes, and with rather simple rules), and the polynomial solutions to NP-complete problems by using an exponential workspace created in a "biological way" (e.g., membrane division, string replication). Other types of formal results are given by normal forms, hierarchies, connections with various formalisms.

In this paper we refer to some "programming" aspects of the membrane systems. We first present an operational semantics of the P systems, together with some correctness results. Then we define several arithmetical operations in membrane systems using a natural encoding of numbers. Finally some software simulators of the membrane systems are presented.

2 Structural Operational Semantics

Membrane systems provide an abstract model for parallel systems, and a suitable framework for distributed and parallel algorithms [6]. For each abstract model, theory of programming introduces various paradigms and uses different notions of computations. Turing machines and register machines are related to imperative programming, and λ-calculus is related to functional programming. It is natural to look at the membrane systems from the point of view of programming theory. This means that we define an abstract syntax, and an operational semantics of the membranes systems. The operational semantics of the membrane systems is given in a big-step style, each step representing the collection of parallel steps due to the maximal parallelism principle. A computation is regarded as a sequence of parallel application of rules in various membranes, followed by a communication step and a dissolving step.

The membrane structure and the multisets in Π determine a configuration of the system. We can pass from a configuration to another one by using the evolution rules. This is done in parallel: all objects, from all membranes, which can be the subject of local evolution rules, as prescribed by the priority relation, should evolve concurrently. Since the right hand side of a rule consists only of messages, an object introduced by a rule cannot evolve at the same step by means of another rule. The use of a rule $u \rightarrow v$ in a region with a multiset w means to subtract the multiset identified by u from w, and then to add the objects of v according to the form of the rule. If an object appears in v in the form $(c, here)$, then it remains in the same region. If we have (c, in_j), then c is introduced in the child membrane with the label j; if a child membrane with the label j does not exist, then the rule cannot be applied. If we have (c, out), then c is introduced in the membrane placed immediately outside the region of the rule $u \rightarrow v$. If the special symbol δ appears in v, then the membrane which delimits the region is dissolved; in this way, all the objects in this region become elements of the region placed immediately outside, while the rules of the dissolved membrane are removed.

Let O be a finite alphabet of objects organized as a free commutative monoid O_c^*, whose elements are called *multisets*. Formally, the set of *membranes for a system* Π, denoted by $\mathcal{M}(\Pi)$, and *the membrane structure* are inductively defined as follows:

- if L is a label, and w is a multiset over $O \cup (O_c^* \times \{here\}) \cup (O_c^+ \times \{out\}) \cup \{\delta\}$, then $\langle L \,|\, w \rangle \in \mathcal{M}(\Pi)$; $\langle L \,|\, w \rangle$ is called *simple (or elementary) membrane*, and it has the structure $\langle\rangle$;
- if $M_1, \ldots, M_n \in \mathcal{M}(\Pi)$ with $n \geq 1$, the structure of M_i is μ_i for all $i \in [n]$, L is a label, w is a multiset over $O \cup (O_c^* \times \{here\}) \cup (O_c^+ \times \{out\}) \cup (O_c^+ \times \{in_{L(M_j)} | j \in [n]\}) \cup \{\delta\}$, then $\langle L \,|\, w \,;\, M_1, \ldots, M_n \rangle \in \mathcal{M}(\Pi)$; $\langle L \,|\, w \,;\, M_1, \ldots, M_n \rangle$ is called *a composite membrane*, and it has the structure $\langle \mu_1, \ldots, \mu_n \rangle$.

A finite set of membranes is usually written as $M_1, \ldots, M_n$. We denote by $\mathcal{M}^+(\Pi)$ the set of non-empty finite sets of membranes. The union of two multisets of membranes $M_+ = M_1, \ldots, M_m$ and $N_+ = N_1, \ldots, N_n$ is written as $M_+, N_+ = M_1, \ldots, M_m, N_1, \ldots, N_n$. An element from $\mathcal{M}^+(\Pi)$ is either a membrane, or a set of sibling membranes.

A *committed configuration* for a membrane system Π is a skin membrane which has no messages and no dissolving symbol δ, i.e., the multisets of all regions are elements in O_c^*. We denote by $\mathcal{C}(\Pi)$ the set of committed configurations for Π, and it is a proper subset of $\mathcal{M}^+(\Pi)$. We have $C \in \mathcal{C}(\Pi)$ iff C is a skin membrane of Π and $w(M)$ is a multiset over O for each membrane M in $\mathcal{C}$.

An *intermediate configuration* is a skin membrane in which we have messages or the dissolving symbol δ. The set of intermediate configurations is denoted by $\mathcal{C}^{\#}(\Pi)$. We have $C \in \mathcal{C}^{\#}(\Pi)$ iff C is a skin membrane of Π such that there is a membrane M in C with $w(M) = w'w''$, $w' \in (Msg(O) \cup \{\delta\})_c^+$, and $w'' \in O_c^*$. By $Msg(O)$ we denote the set $(O^* \times \{here\}) \cup (O^+ \times \{out\}) \cup (O^+ \times \{in_L(M)\})$.

A *configuration* is either a committed configuration or an intermediate configuration. Each membrane system has an initial committed configuration which is characterized by the initial multiset of objects for each membrane and the initial membrane structure of the system.

Each P system has an initial configuration which is characterized by the initial multiset of objects for each membrane and the initial membrane structure of the system. For two configurations C_1 and C_2 of Π, we say that there is a *transition* from C_1 to C_2, and write $C_1 \Rightarrow C_2$, if the following *steps* are executed in the given order:

1. *maximal parallel rewriting step*, consisting of non-deterministically assigning objects to evolution rules in every membrane and executing the rules in a maximal parallel manner;

2. *parallel communication of objects through membranes*, consisting in sending existing messages;

3. *parallel membrane dissolving*, consisting in dissolving the membranes containing δ.

The last two steps take place only if there are messages or δ symbols resulted from the first step, respectively. If the first step is not possible, consequently neither the other two steps, then we say that the system has reached a *halting configuration*. An operational semantics of the P systems, considering each of the three steps, is presented in [2]. We mention here the main results.

We can pass from a configuration to another one by using the evolution rules. This is done in parallel: all objects from all membranes evolve simultaneously according to the evolution rules and their priority relation. The rules of a membrane are using its current objects as much as this is possible in a parallel and non-deterministic way. However, an object produced by a rule cannot evolve at the same step as source of another rule. The use of a rule $u \to v$ in a region with a multiset w has as effect the subtraction of the multiset identified by u from w, followed by the addition of the multiset identified by v.

We denote the *maximal parallel rewriting* on membranes by $\stackrel{mpr}{\Longrightarrow}$ and by $\stackrel{mpr}{\Longrightarrow}_L$ the maximal parallel rewriting over the multisets of objects of the membrane labelled by L (we omit the label whenever it is clear from the context). The rules defining the maximal parallel rewriting use two predicates regarding mpr-irreducibility and (L,w)-consistency.

Proposition 1. Let Π be a membrane system. If $C \in \mathcal{C}(\Pi)$ and $C' \in \mathcal{C}^{\#}(\Pi)$ such that $C \stackrel{mpr}{\Longrightarrow} C'$, then C' is mpr-irreducible.

We denote the *parallel communication relation* by $\stackrel{tar}{\Longrightarrow}$. The rules defining the parallel communication relation use a predicate expressing tar-irreducibility.

Proposition 2. Let Π be a P system. If $C \in \mathcal{C}^{\#}(\Pi)$ with messages and $C \stackrel{tar}{\Longrightarrow} C'$, then C' is tar-irreducible.

We denote the *parallel dissolving relation* by $\stackrel{\delta}{\Longrightarrow}$. The rules defining the parallel dissolving relation use a predicate expressing δ-irreducibility. We note that $C \in \mathcal{C}(\Pi)$ iff C is tar-irreducible and δ-irreducible.

Proposition 3. Let Π be a P system. If $C \in \mathcal{C}^{\#}(\Pi)$ is tar-irreducible and $C \stackrel{\delta}{\Longrightarrow} C'$, then C' is δ-irreducible.

According to the standard description in membrane computing, a *transition step* between two configurations $C, C' \in \mathcal{C}(\Pi)$ is given by: $C \Rightarrow C'$ iff C and C' are related by one of the following relations:

$$\text{either } C \stackrel{mpr}{\Longrightarrow}; \stackrel{tar}{\Longrightarrow} C', \text{ or } C \stackrel{mpr}{\Longrightarrow}; \stackrel{\delta}{\Longrightarrow} C', \text{ or } C \stackrel{mpr}{\Longrightarrow}; \stackrel{tar}{\Longrightarrow}; \stackrel{\delta}{\Longrightarrow} C'.$$

The three alternatives in defining $C \Rightarrow C'$ are given by the existence of messages and dissolving symbols along the system evolution. Starting from a configuration without messages and dissolving symbols, we apply the "mpr" rules and get an intermediate configuration which is mpr-irreducible; if we have messages, then we apply the "tar" rules and get an intermediate configuration which is tar-irreducible; if we have dissolving symbols, then we apply the dissolving rules and get a configuration which is δ-irreducible. If the final configuration has no messages or dissolving symbols, then we say that the transition relation $\Rightarrow$ is well-defined as an evolution between the initial and final configurations.

Proposition 4. The relation $\Rightarrow$ is well-defined over the entire set $\mathcal{C}(\Pi)$ of configurations.

Examples of inference trees, as well as the proofs of the results are presented in [2].

Operational semantics provides us with a formal way to find out which transitions are possible for the current configuration of a membrane system. Given an operational semantics, we can derive easily an interpreter for membrane systems, as well as the basis for the definition of certain equivalences and congruences between membrane systems. Moreover, given an operational semantics, we can reason about the rules defining the semantics. A notion of bisimulation can be defined (see [2]), and the bisimulation relation allows to compare the evolution behaviour of two membrane systems.

3 Arithmetical Operations in Membrane Systems

The problem of number encoding using multisets is interesting and complex. The first paper on the encodings and arithmetical operations in membrane systems is [5]. In [5] we present several combinatorial results and some encodings of numbers using multisets. Here we present some arithmetical operations over numbers encoded by a simple and natural encoding (each object of a membrane represents a unit, and we use n objects to represent the number n). We indicate the complexity of some arithmetical operations, and build the membrane systems which implement the arithmetic operations over the encoded numbers.

Addition
Time complexity: $O(1)$

$$\begin{aligned}\Pi &= (V,\mu,w_0,(R_0,\emptyset),0),\\ V &= \{a,b\},\\ \mu &= [_0]_0,\\ w_0 &= a^n b^m,\\ R_0 &= \{b\rightarrow a\}.\end{aligned}$$

Addition is trivial; we consider n objects a and m objects b. The rule $b\rightarrow a$ says that an object b is transformed in one object a. Such a rule is applied in parallel as many times as possible. Consequently, all objects b are erased. The remaining number of objects a represents the addition $n+m$.

Subtraction
Time complexity: $O(1)$

$$\begin{aligned}\Pi &= (V,\mu,w_0,(R_0,\emptyset),0),\\ V &= \{a,b\},\\ \mu &= [_0]_0,\\ w_0 &= a^n b^m,\\ R_0 &= \{ab\rightarrow\lambda\}.\end{aligned}$$

Subtraction is described in the following way: given n objects a and m objects b, a rule $ab\rightarrow\lambda$ says that one object a and one object b are deleted (this is represented by the empty symbol λ). Consequently, all the pairs ab are erased. The remaining number of objects represents the difference between n and m.

Multiplication without promoters
Time complexity: $O(n\cdot m)$

The object is a promoter for a rule if the rule can be applied only in the presence of object. Figure 1 presents a P system Π_1 without promoters for multiplication of n (objects a) by m (objects b), the result being the number of objects d in membrane 0. In this P system we use the priority relation between rules; for instance $bv\rightarrow dev$ has a

higher priority than $av \to u$, meaning the second rule is applied only when the first one cannot be applied anymore. Initially only the rule $au \to v$ can be applied, generating an object v which activates the rule $bv \to dev$ m times, and then $av \to u$. Now $eu \to dbu$ is applied m times, followed by $au \to v$. The procedure is repeated until no object a is present within the membrane. We note that each time when one object a is consumed, then m objects d are generated.

$$\begin{aligned}
\Pi_1 &= (V, \mu, w_0, (R_0, \rho_0), 0),\\
V &= \{a, b, e, v, u\},\\
\mu &= [_0]_0,\\
w_0 &= a^n b^m u,\\
R_0 &= \{r_1 : au \to v, r_2 : bv \to dev, r_3 : av \to u, r_4 : eu \to dbu\},\\
\rho_0 &= \{r_2 > r_1, r_4 > r_3\}.
\end{aligned}$$

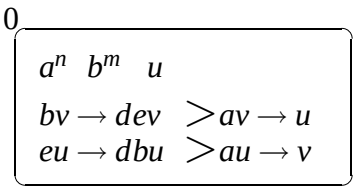


Figure 1: Multiplier without promoters

Multiplication with promoters
Time complexity: $O(n)$

Figure 2 presents a P system Π_2 with promoters for multiplication of n (objects a) by m (objects b), the result being the number of objects d in membrane 0. In this P system we use rules with priority and with promoters. The object a is a promoter in the rule $b \to bd|_a$, i.e., this rule can only be applied in the presence of object a. The available m objects b are used in order to apply m times the rule $b \to bd|_a$ in parallel; based on the priority relation and the availability of a objects (except one a as promoter), the rule $au \to u$ is applied in the same time. The priority relation is motivated because the promoter a is a resource for which the rules $b \to bd|_a$and $au \to u$ are competing. The procedure is repeated until no object a is present within the membrane. We note that each time when one object a is consumed, then m objects d are generated.

$$\begin{aligned}
\Pi_2 &= (V, \mu, w_0, (R_0, \rho_0), 0),\\
V &= \{a, b, u\},\\
\mu &= [_0]_0,\\
w_0 &= a^n b^m u,\\
R_0 &= \{r_1 : b \to bd|_a, r_2 : au \to u\},\\
\rho_0 &= \{r_1 > r_2\}.
\end{aligned}$$

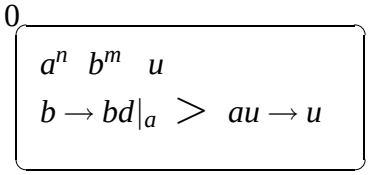


Figure 2: Multiplier with promoters

The membrane systems for multiplication differ from others presented in the literature [16] because they do not have exponential space complexity, and do not require active membranes. As a particular case, it would be

quite easy to compute n^2 by just placing the same number n of objects a and b. Another interesting feature is that the computation may continue after reaching a certain result, and so the system acts as a P transducer [12]. Thus if initially there are n (objects a) and m (objects b), the system evolves and produces $n \cdot m$ objects d. Afterwards, the user can inject more objects a and the system continues the computation obtaining the same result as if the objects a are present from the beginning. For example, if the user wishes to compute $(n+k) \cdot m$, it is enough to inject k objects a at any point of the computation. Therefore this example emphasizes the asynchronous feature and a certain degree of reusability and robustness.

Division

We implement division as repeated subtraction. We compute the quotient and the remainder of n_2 (objects a in membrane 1) divided by n_1 (objects a in membrane 0) in the same P system evolution. The evolution starts in the outer membrane by applying the rule $a \to b(v, in_1)$. The (v, in_1) notation means that the object v is injected into the child membrane 1. Therefore the rule $a \to b(v, in_1)$ is applied $n_1$times converting the objects a into objects b, and object v is injected in the inner membrane 1. The evolution continues with a subtraction step in the inner membrane, with the rule $av \to e$ applied n_1 times whenever possible.

$$
\begin{aligned}
\Pi &= (V, \mu, w_0, w_1, (R_0, \rho_0), (R_1, \rho_1), 0),\\
V &= \{a, b, b', c, s, u, v\},\\
\mu &= [_0[_1]_1]_0,\\
w_0 &= a^{n_1}s,\\
w_1 &= a^{n_2}s,\\
R_0 &= \{a \to b(v, in_1), b' \to a, r_1 : bu \to b'|_{\neg v}, r_2 : u \to \delta|_{\neg v}, r_3 : csu \to u|_v\},\\
\rho_0 &= \{r_1 > r_2, r_2 > r_3\},\\
R_1 &= \{r_1 : av \to e, r_2 : v \to (v, out),\\
&\quad\ r_3 : es \to s(u, out)(c, out), r_4 : e \to (u, out)\},\\
\rho_1 &= \{r_1 > r_2, r_2 > r_3, r_3 > r_4\}.
\end{aligned}
$$

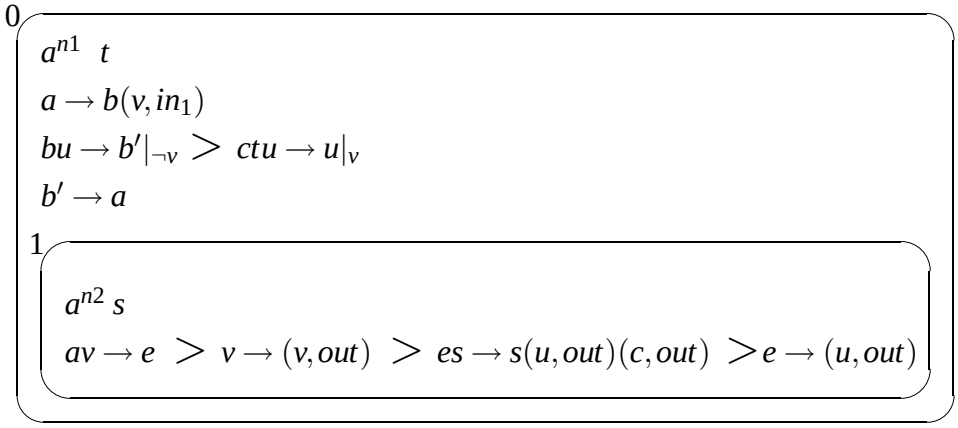


Figure 3: P system for division

Two cases are distinguished in the inner membrane:

- If there are more objects a than objects v, only the rules $es \to s(u, out)(c, out)$ and $e \to (u, out)$ are applicable. Rule $es \to s(u, out)(c, out)$ sends out to membrane 0 a single c (restricted by the existence of a single s into this membrane) for each subtraction step. The number of objects c represents the quotient. On the other hand, both rules send out n_1 objects u (equal to the number of objects e). The evolution continues in the outer membrane by applying $bu \to b'|_{\neg v}$ of $n_1$times, meaning the objects b are converted into objects b' by consuming the objects u only in the absence of v ($|_{\neg v}$ denotes an inhibitor having an effect opposite to that of a promoter). Then the rule $b' \to a$ produces the necessary objects a to repeat the entire procedure.

- When there are less objects a than objects v in the inner membrane we get a division remainder. After applying the rule $av \rightarrow e$, the remaining objects v activate the rule $v \rightarrow (v, out)$. Therefore all these objects v are sent out to the parent membrane 0, and the rules $es \rightarrow s(u, out)(c, out)$ and $e \rightarrow (u, out)$ are applied. Due to the fact that we have objects v in membrane 0, the rule $bu \rightarrow b'|_{\neg v}$ cannot be applied. Since n_2 is not divisible by n_1, the number of the left objects u in membrane 0 represents the remainder of the division. A final cleanup is required in this case, because an object c is sent out even if we have not a "complete" subtraction step; the rule $ctu \rightarrow u|_v$ removes that extra c from membrane 0 in the presence of v. This rule is applied only once because we have a unique t in this membrane.

The natural encoding is easy to understand and work with. However it has the disadvantage that the membranes can contain a very large number of objects when working with very large numbers. We introduce and study the most compact encoding using two object types (binary case) in [5], where we present other P systems implementing the arithmetical operations on numbers encoded using the binary cases of the most compact encoding. We use a web-based simulator available at `http://psystems.ieat.ro` to implement the arithmetical operations, and test each P system.

4 Software Implementations

Several programming paradigms and programming languages have been selected for implementing membrane systems simulators: Lisp, Haskell, MzScheme (as functional programming languages) Prolog, CLIPS (as declarative languages), C, C++, Java (as imperative and object-oriented languages). The user interface can be designed separately from the engine performing the evolution, and it is possible to use different programming languages able to communicate with each other. Each programming paradigm, each programming language has advantages and disadvantages.

Transition membrane systems and deterministic membrane systems with active membranes are simulated in Prolog [14]; they are used to solve NP-complete problems as SAT, VALIDITY, Subset Sum, Knapsack, and partition problems. Sevilla carpets describing the complexity of a membrane system computation [11] are used as a graphical representation for a partition problem in [20].

Membrane systems with active membranes, input membrane and external output are simulated in CLIPS and used to solve NP-complete problems in [18]. The simulator presented in [18] allows to observe the evolution of the systems with active membranes based on production system techniques. The set of rules and the configurations in each step of the evolution are expressed as facts in a knowledge base.

Rewriting membrane systems and membrane systems with symport/antiport rules are described as executable specifications in MAUDE in [2]. The advantage of this approach is that it uses the existing tools of Maude, and it is used to verify the temporal properties of the membrane systems expressed in linear temporal logic.

A more complex simulator (written in Visual C++) for membrane systems with active membranes and catalytic membrane systems is presented in [10]. It provides a graphical simulator, interactive definition, visualization of a defined membrane system, a scalable graphical representation of the computation, and step-by-step observations of the membrane system behaviour. The simulation of these membrane systems has to deal with the potential growth of the membrane structure and adapt dynamically the topology of the configurations depending if some membranes are added or deleted. Polynomial-time solutions to **NP**-complete problems via membrane systems can be reached trading time by space. This is done by producing (via membrane division) an exponential amount of membranes that can work in parallel.

In [10] it is presented a software implementation which provides a graphical simulation for two variants of membrane systems: for the initial version of catalytic hierarchical cell systems, and for membrane systems with active membranes. Its main functions are given by an interactive definition of a membrane system, a visualization of a defined membrane system, a graphical representation of the computation and final result, and saving and (re)loading a defined membrane system. The application is implemented in Microsoft Visual C++ using MFC classes. For a scalable graphical representation, the Microsoft DirectX technology is used. One of the main features of this technology is that the size of each component of the graphical representation is adjusted according to the number of membranes of the system. The system is presented to the user with a graphical interface where the main screen is divided into two windows: The left window gives a tree representation of the membrane system including objects and membranes. The right window provides a graphical representation of the membrane system given by Venn-like diagrams. A menu allows the specification of a membrane system for adding new objects, membranes, rules and priorities. By using the functions *Start*, *Next* and *Stop*, the users can observe the system

evolution step-by-step.

By simulating parallelism and nondeterminism on a sequential machine one can lose the power and attractiveness of membrane system computing. Parallel and cluster implementation for transition membrane systems in C++ and MPI are reported in [8] and [9]. The rules are implemented as threads. At the initialization phase, one thread is created for each rule. Rule applications are performed in terms of rounds. To synchronize each thread (rule) within the system, two barriers implemented as mutexes are associated with the thread. At the beginning of each round, the barrier that the rule thread is waiting on is released by the primary controlling thread. After the rule application is done, the thread waits for the second barrier, and the primary thread locks the first barrier. Since each rule is modelled as a separate thread, it should have the ability to decide its own applicability in a particular round. Generally speaking, a rule can run when no other rule with higher priority is running, and the resources required are available. When more than one rule can be applied in the same conditions, the simulator picks randomly one among the candidates. With respect to the synchronization and communication, for every membrane, the main communication is done by sending and receiving messages to and from its father and children at the end of every round. With respect to the termination, when the system is no longer active, there is no rule in any membrane that is applicable. When this happens, the designated output membrane prints out the result and the whole system halts. In order to detect if the membrane system halts, each membrane must inform the other membranes about its inactivity. It can do so by sending messages to others, and by using a termination detection algorithm [4].

The implementation was designed for a cluster of computers. It is written in C++ and it makes use of *Message Passing Interface (MPI)* as its communication mechanism. MPI is a standard library developed for writing portable message passing applications, and it is implemented both on shared-memory and on distributed-memory parallel computers. The program was implemented and tested on a Linux cluster at the National University of Singapore; the cluster consisted of 64 dual processor nodes.

The above implementations represent the first generation of membrane systems simulators. The recent developments are related to biological applications, and to a new generation of Web-based simulators. WebPS is an open-source web-enabled simulator for membrane systems [6]. The simulator is based on CLIPS, and it is already available as a Web application. As any Web application, WebPS does not require an installation. It can be used from any machine anywhere in the world, without any previous preparation. A simple and easy to use interface allows the user to supply an XML input both as text and as a file. A friendly way of describing membrane systems is given by an interactive JavaScript-based membrane system designer. The interface provides a high degree of (re)usability during the development and simulation of the membrane systems. The initial screen offers an example, and the user may find useful documentation about the XML schema, the rules, and the query language. The query language helps the user to select the output of the simulation. The simulator is free software, and it offered at `http://psystems.ieat.ro` under the *GNU General Public License*. This allows anyone to contribute with enhancements and error corrections to the code, and possibly develop new interfaces for the C and CLIPS level APIs. These interfaces can be local (graphical or command-line), or yet other Web-based ones.

In the same paper [6], the authors present an accelerator for parallelization of the existing sequential simulators. This accelerator is used to parallelize an existing CLIPS simulator [18]. The speedup and the efficiency of the resulting parallel implementation are surprisingly close to the ideal ones.

5 Conclusion and Related Work

Structural operational semantics is an approach originally introduced by Plotkin [19] in which the operational semantics of a programming language or a computational model is specified in a logical way, independent of a machine architecture or implementation details, by means of rules that provide an inductive definition based on the elementary structures of the language or model. Structural operational semantics is intuitive and flexible, and it becomes more attractive during the years by the developments presented by Kahn [15] and Milner [16]. Configurations are states of transition systems, and computations consists of sequences of transitions between configurations, and terminating (if it terminates) in a final configuration. We present a structural operational semantics of the membrane systems; the inference rules provide a big-step operational semantics due to the parallel nature of the model. A structural operational semantics of the systems emphasizes also the deductive nature of the membrane computing by describing the transition steps by using a set of inference rules. Considering a set $\mathcal{R}$ of inference rules, we can describe the computation of a membrane system as a deduction tree. In [3] we translate the big-step operational semantics of membrane systems into rewriting logic. By using the rewriting engine Maude [13], we obtain an interpreter for membrane systems, and verify various properties of these systems.

Looking at the membrane systems from the point of view of programming theory, we define an appropriate

data representation for P systems, and make the first steps to define an arithmetic unit for these abstract machine inspired by cells. The natural encoding over multisets is very close to biology, and can help to understand some biological mechanisms, improving also some computational models inspired by biology.

We have designed and implemented sequential and parallel software simulators; we present some of them, and compare with other software simulators of the P systems. A web-based implementation is presented in [6].

Acknowledgements

The contributions of this paper were obtained together with my colleagues. Many thanks to Oana Andrei and Dorel Lucanu for the joint work on the operational semantics of the membrane systems. Many thanks to Cosmin Bonchiş and Cornel Izbaşa for their contributions to the arithmetical operations over multisets in the framework of membrane systems, and to the software implementation WebPS.

References

[1] O. Andrei, G. Ciobanu, D. Lucanu. Executable Specifications of the P Systems. In *Membrane Computing WMC5*, LNCS vol.3365, Springer, 127-146, 2005.

[2] O. Andrei, G. Ciobanu, D. Lucanu. Structural Operational Semantics of P Systems. *Proceedings WMC6*, LNCS vol.3850, Springer, 32-49, 2006.

[3] O. Andrei, G. Ciobanu, D. Lucanu. Operational Semantics and Rewriting Logic in Membrane Computing. *Proceedings SOS Workshop* 2005, to appear in *ENTCS*.

[4] H. Attiya, J. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. McGraw-Hill, 2000.

[5] C. Bonchiş, G. Ciobanu, C. Izbaşa. Encodings and Arithmetic Operations in Membrane Computing. In Jin-Yi Cai, S. Barry Cooper, Angsheng Li (Eds.): *Theory and Applications of Models of Computation*, LNCS 3959, Springer, 618–627, 2006.

[6] C. Bonchiş, G.Ciobanu, C. Izbaşa, D. Petcu. A Web-based P systems simulator and its parallelization. In C.Calude et al. (Eds.): *Unconventional Computing*, LNCS vol.3699, Springer, 58-69, 2005.

[7] G. Ciobanu. Distributed Algorithms over Communicating Membrane Systems. *Biosystems* vol.70, Elsevier, 123-133, 2003.

[8] G. Ciobanu, R. Desai, A. Kumar. Membrane Systems and Distributed Computing. In *Proceedings WMC3*, LNCS vol.2597, Springer, 187-202, 2003.

[9] G. Ciobanu, W. Guo. P Systems Running on a Cluster of Computers. In *Procedings 4th WMC*, Taragona, LNCS vol.2933, Springer, 123-139, 2004.

[10] G. Ciobanu, D. Paraschiv. P System Software Simulator. *Fundamenta Informaticae* 49, 61-66, 2002.

[11] G. Ciobanu, Gh. Păun, Gh.Ştefănescu. Sevilla Carpets Associated with P Systems. *Report 26/03 Rovira i Virgili University*, Tarragona, 135-140, 2003.

[12] G. Ciobanu, Gh. Păun, Gh. Ştefănescu. P Transducers. *New Generation Computing* **24**, 1–28, 2006.

[13] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada. Maude: Specification and Programming in Rewriting Logic. *Theoretical Computer Science*, vol.285, 187-243, 2002.

[14] A. Cordon-Franco, M.A. Gutierrez-Naranjo, M.J. Perez-Jimenez, A. Riscos-Nunez, F. Sancho-Caparrini. Implementing in Prolog an Effective Cellular Solution for the Knapsack Problem. In *Proceedings 4th WMC*, Taragona, LNCS vol.2933, Springer, 140-152, 2004.

[15] G. Kahn. *Natural semantics*, Technical Report 601, INRIA Sophia Antipolis, 1987.

[16] R. Milner. Operational and algebraic semantics of concurrent processes. In J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science* vol.B, 1201-1242, Elsevier Science, 1990.

[17] Gh. Păun. *Membrane Computing. An Introduction*. Springer, 2002.

[18] M.J. Perez-Jimenez, F.J. Romero-Campero. A CLIPS Simulator for Recognizer P Systems with Active Membranes. In *Proceedings 2nd Brainstorming Week on Membrane Computing*, University of Sevilla Tech. Rep 01/2004, 387-413, 2004.

[19] G. Plotkin. Structural operational semantics. *Journal of Logic and Algebraic Programming* vol.60, 17-139, 2004.

[20] A. Riscos-Núñez, *Cellular Programming: Efficient Resolution of Numerical NP-Complete Problems*. PhD Thesis, University of Seville, 2004.

Gabriel Ciobanu
Romanian Academy
Institute of Computer Science
Address: Blvd. Carol I nr.8, Iaşi
E-mail: gabriel@iit.tuiasi.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 23-30

Pathogen Variability. A Genomic Signal Approach

Invited paper

Paul Dan Cristea

Abstract: The conversion of genomic symbolic sequences into digital signals has been applied for the analysis pathogen variability. Results are given on the variability of Human Immunodeficiency Virus, type 1, subtype F, isolated in Romania, and of the type A avian influenza virus H5N1, for which sequences have been downloaded from GenBank [1]. Nucleotide sequence analysis is corroborated with techniques based on the genomic signal approach to detect pathogen resistance to antiretroviral treatment. In the case of protease (PR) inhibitors, it is found that the treatment induces single nucleotide polimorphisms (SNPs) in specific sites. For moderate resistance, the changes affect the PR enzyme only at the level of the protein, whereas for multiple drug resistance, the RNA gene secondary structure also changes.

Keywords: Genomic signals, Pathogen variability, HIV, Influenza, Orthomyxoviridae, Drug resistance

1 Introduction

As shown in a series of previous papers [2-4], the conversion of nucleotide and amino acid sequences into digital signals offers the possibility to apply signal processing methods for the analysis of genomic data. The genomic signal conversion used in our work is a one-to-one mapping of symbolic genomic sequences into complex signals, as described in [2]. The idea is to conserve all the information in the initial symbolic sequence, while bringing in foreground some features significant for the subsequent processing and analysis. This direct method has proven its potential in revealing large scale features of DNA sequences, maintained at the scale of whole genomes or chromosomes, including both coding and non-coding regions. One of the most conspicuous results is that the unwrapped phase of DNA complex genomic signals varies almost linearly along all investigated chromosomes, for both prokaryotes and eukaryotes. The slope is specific for various taxa and chromosomes. This regularity of the genomic signals reveals a corresponding large scale regularity in the distribution of pairs of successive nucleotides, which is similar to Chargaff's first order rules for the frequencies of occurrence of the nucleotides [5].

We applied the same genomic signal approach for studying the variability of several pathogens, including the Human Immunodeficiency Virus, type 1 (HIV-1), subtype F, isolated from Romanian patients at the National Institute of Infectious Diseases "Prof. Dr. Matei Bals", Bucharest [3], and the avian influenza virus type A, based on genomic sequences downloaded from GenBank [1].We have used mainly the phase analysis of the complex genomic signals attached to the nucleotide sequences describing viral genes, as well as the analysis of the corresponding secondary RNA structure and of the phylogenetic neighbor-joining trees for some of these genes.

The focus of the study is primarily on the enzyme changes involved in generating pathogen resistance to multiple drug treatment. A novel methodology for describing sets of related genomic signals, based on a common reference and on individual differences has been developed. Variability signals with respect to average, median and maximum flat references, and digital derivatives of genomic signals are applied to this purpose. Applying this method, it has been found that the mutations in the genes of the analyzed viruses occur only in some specific, well defined locations, while the largest part of their genome remains unchanged. The mutations conferring drug resistance are a subset of all mutations occurring in the studied viruses.

On the other hand, for the case of HIV protease, it has been shown that the changes in response to the antiretroviral drug treatment occur not only at the level of the final enzyme product, preventing the action of the drug on the active protease catalytic site, but also at the level of protease gene RNA secondary structure. These type of changes have been found only for multiple drug resistant viruses.

2 Symbolic Sequence Conversion

For convenience we repeat here the mapping used in our work for the representation of the nucleotides [2]

$$a = 1 + j, c = -1 - j, g = -1 + j, t = 1 - j \tag{1}$$

Apart of the mapping of the four nucleotides (a,c,g,t), the complete genomic signal representation of nucleotide sequences also comprises the mapping of all the other IUPAC symbols for nucleotide classes: $s=\{c,g\}$ - strongly bonded, $w=\{a,t\}$ - weakly bonded, $r=\{a,g\}$ - purines, $y=\{c,t\}$ - pyrimidines, $m=\{a,c\}$ - amine, $k=\{g,t\}$ - ketone, $b=\{c,g,t\}=\neg a$, $d=\{a,g,t\}=\neg c$, $h=\{a,c,t\}=\neg g$, $v=\{a,c,g\}=\neg t$, and $n=\{a,c,g,t\}$ [2]. These symbols occur in the nucleotide sequences generated by genotyping because of the multiplicities determined either by the variability within the virus population or by noise. But this is not the case of the consensus sequences downloaded from GenBank [1], which are curated to contain only the (a,c,g,t) nucleotide symbols. The mapping in equation (1) has the advantage of conserving all the information in the initial symbolic sequence, as it uses a bijective mapping, while being as little biased as possible.

3 Representation by Reference and Variation

To study the variability of the genomic signals in a given set, for example, the signals for multiple resistant viruses, it is convenient to use a description comprising two types of components: (1) the reference - a certain signal considered to best describe the common variation of all components in the considered cluster; (2) the difference of each signal in the cluster with respect to the common reference. In such an approach, it is important to introduce in the common reference as much as possible of the variation shared by all the signals, and keep for the individual differences of each signal only the variations belonging actually to the that signal, without external variation.

The reference can be chosen as one of the following possibilities:

- average (mean) of the signals, or another linear combination of the signals;
- median - the signal in the central position, or the average of the pair of signals placed centrally;
- maximum flat signal - a modified median that keeps better local variations on the signals where they occur avoiding spurious transfers on other signals.

When the reference equals the average, the dispersion of the cluster of signals is minimum, i.e., the sum of the squares of the individual differences between each signal and the reference is minimized. But the average, as any other linear combination, has the important disadvantage that a localized variation of only one of the signals is transmitted to the reference, so that all the other signals will have an apparent variation of opposite sign in that point.

The median reference performs better, being a nonlinear function of the signals in the cluster, so that it decouples the common reference from the local variations of each of the individual signals. The median reference minimizes the sum of the absolute values of the differences between each signal and the reference. A variation localized on only one of the signals is no longer transmitted to the reference, so that it does not affect the variation with respect to the reference of the other signals. The exception occurs when the signal on which the localized variation occurs is just the median.

The maximum flat (MaxFlat) reference is equal to the median wherever the median has no variations which are not shared by other signals. Elsewhere, the MaxFlat reference assumes the minimal variation that corresponds to its trend, if possible remaining constant. Consequently, the variation signals show better the changes that occur in each individual signal, with less "crosstalk". The digital derivatives of the variation signals show only the actual changes, caused by the variability in each of the signals and, for genomic signals, correspond directly to the SNPs.

4 HIV-1 Subtype F Variability

A phase analysis has been performed on a segment of about 1302 base pairs, approximately aligning with the standard sequence of HIV-1 (NC001802) in GenBank [1] over the interval 1799..2430 bp. This segment, which is currently used for the standard identification and assessment of HIV-1 strains, comprises the protease (PR) gene and almost two thirds of the reverse transcriptase (RT) gene. The PR and RT segments are contiguous and have been analyzed both together, as one entity, and independently, as two distinct encoding regions. The PR gene has the length 297 bp and is located in the first interval (1..297 bp) of the sequenced DNA segment, respectively along the 1799..2095 bp region of the NC001802 sequence. The RT encoding segment that has been analyzed has a length of 1005 bp and is located in the second interval (298..1302 bp) of the analyzed DNA segment, respectively along the 2096..3100 bp region of the NC001802 sequence. The entire RT gene has 1680 bp located in the interval 2096..3775 of the sequence.

Figures 1 and 2 show the cumulated and unwrapped phase of genomic signals for the protease (PR) genes from nine instances of HIV type 1, F clade [1, 6]. Three cases come from treatment naïve patients (S - sensitive), three from patients that developed resistance to one of the drugs (R), and three with multiple resistance to ther antiretroviral treatment (M). The cumulated phase is proportional to the unbalance in the number of nucleotides (statistics of first order) along the nucleic acid strand given by: $3(n_G - n_C) + (n_A - n_T)$, up to a $\pi/4$ factor, whereas the unwrapped phase is proportional to the difference between the number of direct and inverse nucleotide transitions (statistics of second order) along the nucleic acid strand $(n_+ - n_-)$, with a $\pi/2$ factor [2]. Figures 3 and 4 give the same informtion for the segment comprising 1005 bp of reverse transcriptase (RT) genes, out of the total of 1680 bp in this gene, for the same isolates in Figs. 1 and 2. As expected, the cumulated phase varies less than the unwrapped phase for these instances, as all mutations are of the SNP type and affect more the nucleotide pair distribution than the nucleotide distribution itself. Even for the unwrapped phase, the variation of the signal along the strand is quite similar for most of the sequences, but the local changes cumulate along the strands. Because of the mutations are local, the general shape of the phase signals are similar. It is also to be noticed that all the genomic material in these sequences is encoding and uses the same reading frame.

The vertical strips in these figures mark the positions of the mutations (SNPs) that induce resistance to protease inhibitors (Indinavir, Ritonavir, Saquinavir, Nelfinavir, and Amprenavir) [1]. The mutations that lead to multiple drug resistance are concentrated in several sites. In most of the remaining genome, the viruses have the same longitudinal structure. The sequences display mutations in several other locations. The effect of the mutations can easier be seen on the unwrapped phase, which is more sensitive to SNPs.

The successive mutations of the SNP type do not induce the divergence that could be expected, so that the signals do not actually diverge from one another. On the contrary, the signals tend to cluster, as the variations tend to compensate each other, so that the overall span of the signals does not increase directly with the number of mutations and the number of signals. This is another proof of the fact that, from the structural point of view, a genomic sequence satisfies more restrictions than a "plain text", which must just correspond to a certain semantics and to certain grammar rules, and resembles more to a “poem”, which additionally obeys rules of symmetry, giving its “rhythm” and “rhyme”. The recurrence of such patterned structures is reflected in simple mathematical rules satisfied by the corresponding genomic signals.

The representation can be improved by using the reference-difference description, choosing the maximum flat (MaxFlat) reference, as shown in Fig. 5 for the unwrapped phase in Fig. 4. In this case, the largest possible part of the common behavior of the signals is introduced in the reference signal, whereas each individual variation signal maintains only the changes occurring in that particular signal, or to the class it belongs to. The reference signal is no longer necessarily equal in each interval with one of the signals, even when the number of signals is odd. The digital derivatives of the difference signals, shown in Fig. 6 show only the actual changes caused by the variability in each of the signals. In the case of HIV, these changes correspond directly to the SNPs. For multiple resistant strains, the pulses correspond to the sites known from literature to confer resistance to various drugs.

HIV-1 makes many of its proteins in one long chain, and protease (PR) has the essential role of cutting this ’polyprotein’ into the proper pieces, with the proper timing. Consequently, PR has been chosen as an important target for the current drug anti-HIV therapy. PR is a small enzyme, comprising two identical peptide chains, each of 99 amino acids long, which are encoded by the same gene of 297 nucleotides.

The two chains form a tunnel that holds the polyprotein, which is cut at an active site located in the center of the tunnel. Drugs bind to PR, blocking its action. Studying the estimated secondary structure of the PR RNA for the nine virions previously analyzed, it can be shown [3] that the structures are quite similar for drug sensitive and drug simple resistant viruses. This result is consistent with the generally accepted model stating that the genomic changes of HIV, which induce resistance to drugs, operates at the level of the protein (the final protease enzime), preventing the blocking of its catalytic site. On the other hand, it is found the remarkable fact that, for drug multiple resistant strains, there is a significant change in the RNA secondary structure. Large loops and bulges are replaced with similar, but smaller, less vulnerable, closed-loop structures. These results indicate that there is a certain action of the drug at the level of the protease RNA, effect that becomes evident when mutations conferring multiple drug resistance occur.

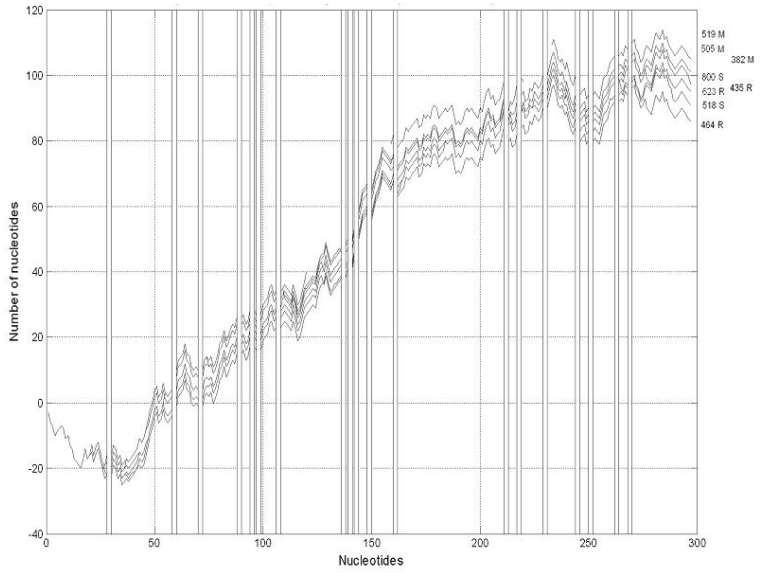


Figure 1: Cumulated phase expressed by $3(n_G - n_C) + (n_A - n_T)$ [2] for the protease (PR) gene of nine isolates of HIV-1, subtype F, showing sensitivity (S), resistance (R) and multiple resistance (M) to drugs.

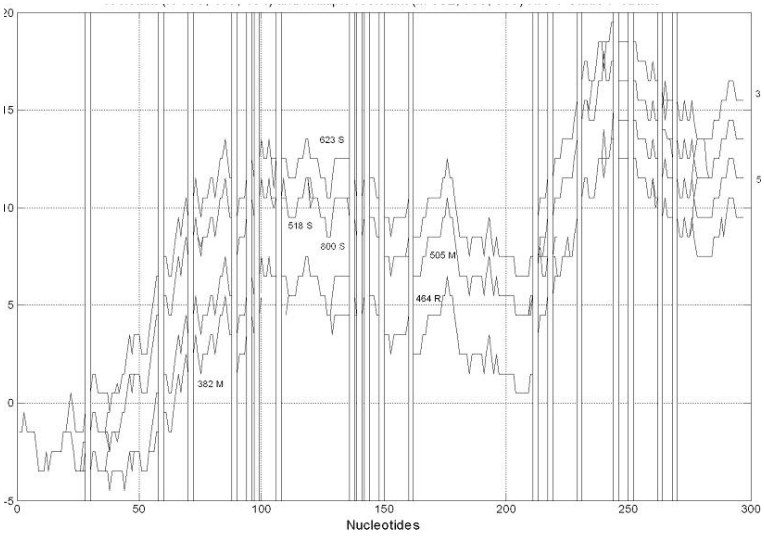


Figure 2: Unwrapped phase expressed by $n_+ - n_-$ [2] for the protease gene of the isolates of HIV-1 in Fig. 1.

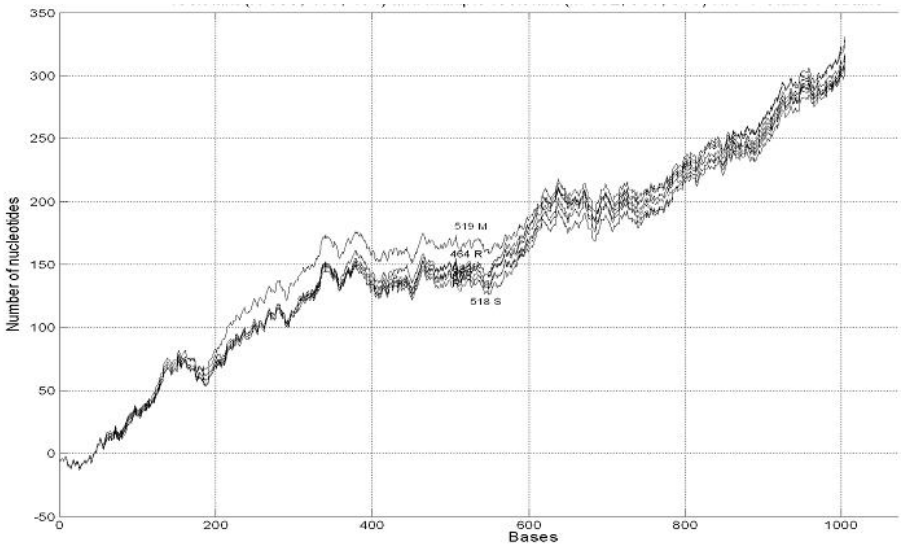


Figure 3: Cumulated phase of RT genomic signals for the isolates shown in Figs. 1 and 2.

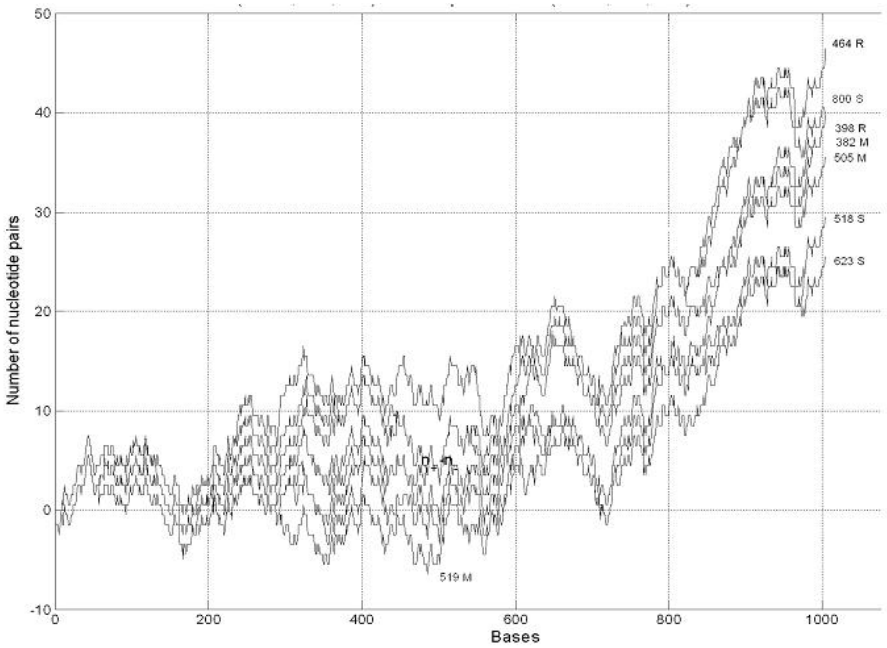


Figure 4: Unwrapped phase for the RT gene in the isolates shown in Figs. 1 and 2.

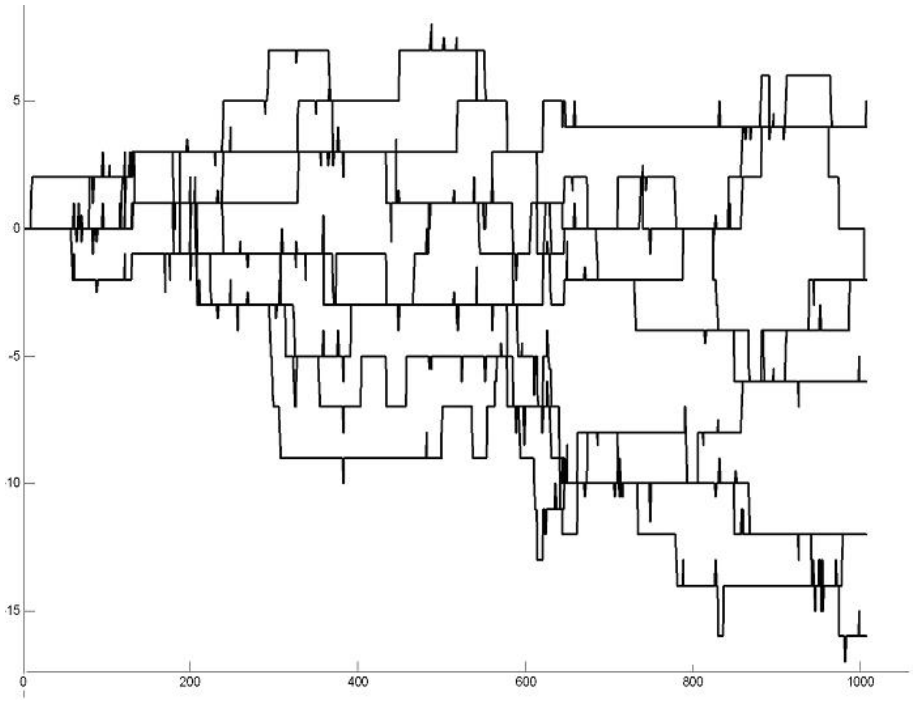


Figure 5: The unwrapped phase in Fig. 4 shown with respect to the *MaxFlat* reference.

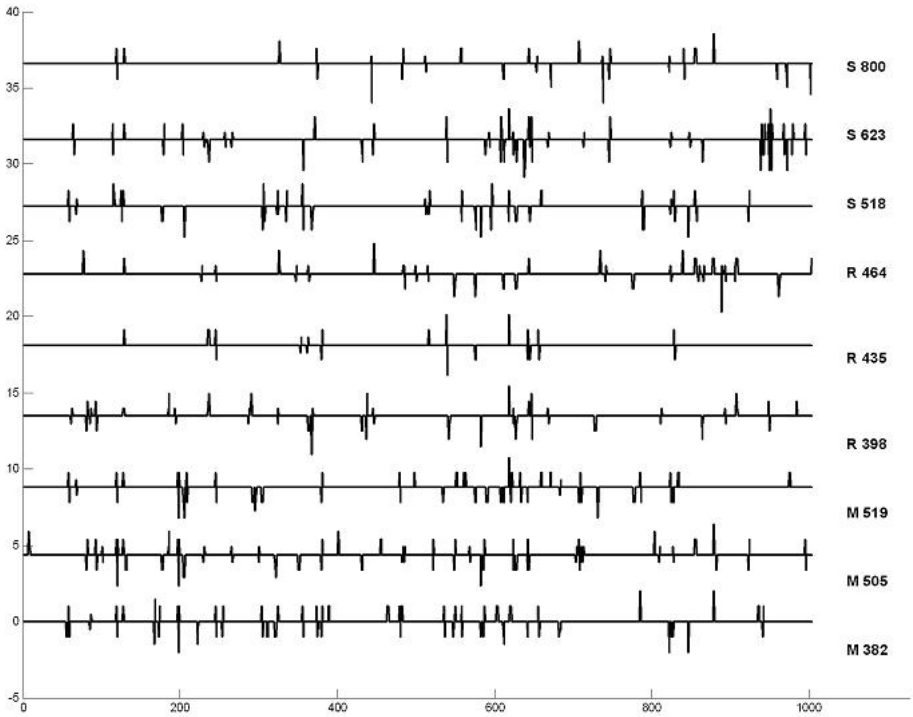


Figure 6: Digital derivatives of the variation signals in Fig. 5.

5 Variability of Hemagglutinin gene of influenza H5N1 virus

The influenza virus envelope embeds two specific antigenic glycoproteins that project out of the virion surface, the *Hemagglutinin* (HA) and the *Neuraminidase* (NA). Many different combinations of HA and NA proteins are possible, but only the H1N1 (Spanish endemic), H1N2 (Asian epidemic), and H3N2 (Hong Kong epidemic) subtypes have circulated worldwide among humans. HA protein selectively binds to the sialic acid of the host cell surface receptors, thus recognizing the cells that the virus can invade [4, 6]. Figure 7 gives the cumulated phase of the HA gene for H5N1 viruses isolated from two humans (AF046080, AF046097) and one chicken (AF046088), in Hong Kong, in 1997 [6, 7]. The genes for viruses isolated close in time are similar, even when crossing the inter-species barrier, whereas a large variation can be seen for genes isolated at larger time intervals. Only several SNPs are found in Fig. 8 which gives the difference cumulated phases with respect to the MaxFlat reference. The same result has been obtained for all the genes in the eight segments of the H5N1 virus [4, 6].

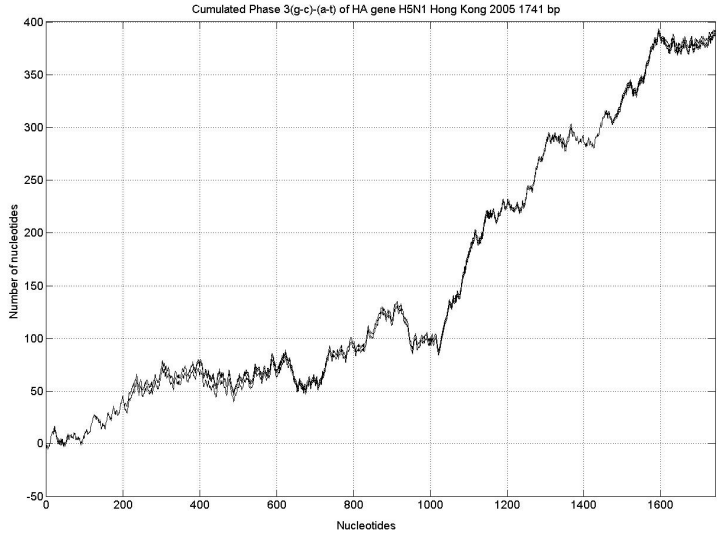


Figure 7: Cumulated phase of the HA gene, H5N1 virus (accessions AF046080, 88, 97 [1, 6]).

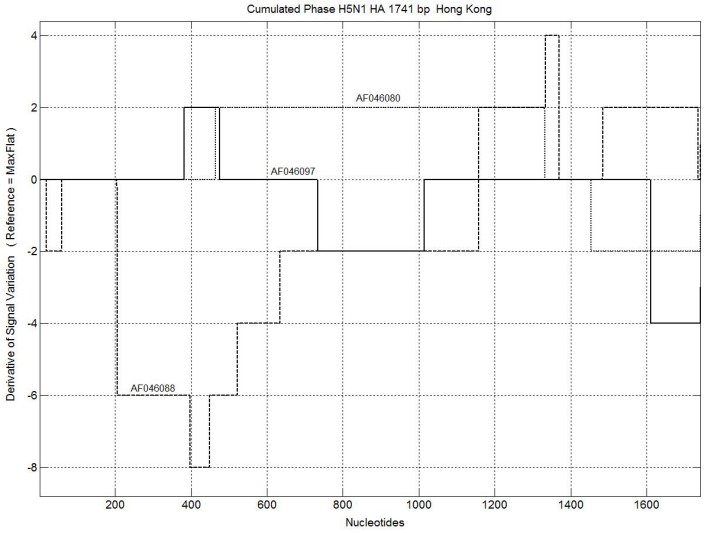


Figure 8: Differences of HA gene cumulated phases in Fig.7 with respect to the MaxFlat reference.

6 Further Work

Further work will be focused on:

- the dynamics of Influenza Type A viruses that have crossed till now the species barrier from birds to humans, and which hold the potential to become highly contagious and highly lethal in humans, including the H5N1 subtype,
- extending the study from the nucleotide to the amino acid level, which could be more significant from the phenotypic point of view,
- using genomic signals for helping clustering viruses in classes.

Acknowledgments

The sequences of HIV presented in this paper have been genotyped by Dr. Dan Otelea from the National Institute of Infectious Diseases “Prof. Dr. Matei Bals”, Bucharest, Romania. Results referring to the study of HIV variability have been previously jointly published [3].

References

[1] National Center for Biotechnology Information, National Institutes of Health, National Library of Medicine, National Center for Biotechnology Information, GenBank, http://www.ncbi.nlm.nih.gov/genoms

[2] P. D. Cristea, "Representation and Analysis of DNA sequences", in Genomic Signal Processing and Statistics, Editors E.G. Dougherty, I. Shmulevici, Jie Chen, Z. J. Wang, Book Series on Signal Processing and Communications, Hidawi, 2005, pp.15-65.

[3] P. D. Cristea, D. Otelea, Rodica Tuduce, "Study of HIV Variability Based on Genomic Signal Analysis of Protease and Reverse Transcriptase Genes", EMBC’05, Sept. 2005, Shanghai, China.

[4] P. D. Cristea, "Genomic Signal Analysis of Pathogen Variability", SPIE, BO24, paper 5699-52, San Jose, Jan, 2005, 12 pg.

[5] E. Chargaff, "Structure and function of nucleic acids as cell constituents", Fed. Proc., 10, pp. 654-659, 1951.

[6] D.L. Suarez,.et.al., Comparisons of highly virulent H5N1 influenza A viruses isolated from humans and chickens from Hong Kong, J. Virology, vol. 72 (8), pp. 6678-6688, 1998. (AF046080-99).

[7] E. Ghedin, N. Sengamalay, M. Shumway et. al., "Large-scale sequencing of human influenza reveals the dynamic nature of viral genome evolution", Nature, vol. 437, Oct. 2005, pp.1162-1166.

[8] M.S. Hirsch et al., "Antiretroviral Drug Resistance Testing in Adult HIV-1 Infection", in Proc. Recommendations of an International AIDS Society - USA Panel, JAMA, vol. 283, no. 18, May 10, 2000, pp.2417-2426.

Paul Dan Cristea
University POLITEHNICA of Bucharest
Biomedical Engineering Center
Address: Spl. Independentei 313, sect. 6
060042 Bucharest, Romania
E-mail: pcristea@dsp.pub.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 31-39

Evolutionary Optimization of Coercive Functionals Defined on Euler-Monge Surfaces with Fixed Boundary Curves

Invited paper

Dan Dumitrescu, Ágoston Róth

Abstract: A new evolutionary multimodal optimization technique is applied for coercive functionals defined on a set of functions of the type $u \in C^2(\Omega, \mathbb{R})$, where Ω is a non-empty compact subset of $\mathbb{R}^2$. u defines an Euler-Monge surface, which has fixed boundary curves. In our approach these boundaries are approximated by a C^2-continuous curve.

Keywords: evolutionary multimodal optimization, roaming technique, coercive energy functionals

1 Introduction

Several optimization problems in engineering and mechanics can be reduced to find equilibrium/critical points of certain functionals defined on a set of functions of the type $u : \Omega \to \mathbb{R}$, where Ω is a non-empty compact subset of $\mathbb{R}^2$.

Such functionals appear at adhesively connected plates, at sandwich beams with double buckling load, at sandwich Timoshenko plates, at cylindrical shells and their buckling or at laminated von Kármán plates (see [2][3]). Not only the study of the existence, but also the *multiplicity* and *localization* of the critical points of such functionals (in particular, *global* or *local minima*) constitutes a novel and challenging problem.

Our aim is to develop an evolutionary technique for detecting *all* global or local optima of such functionals. We emphasize that more complicated optimization problems can be handled by the proposed approach (for instance, constrained optimization problems).

2 Prerequisites

Let us denote by Ω a compact and non-empty domain of $\mathbb{R}^2$ having one hole inside. Consider a (not necessarily) continuous function $\psi : \mathbb{R} \to \mathbb{R}$, fulfilling the growth condition

$$|\psi(t)| \leq c(1+|t|^s), \forall t \in \mathbb{R}, \tag{1}$$

with fixed $s \in]0,1[$ and $c > 0$. Let us introduce the notation

$$\mathcal{H}_0(\Omega) = \left\{ u : \Omega \to \mathbb{R} |\ u_{|\partial\Omega} \equiv 0 \right\}.$$

Several real-world engineering or mechanical problems lead to solving differential equations of the type

$$\left\{ \begin{array}{rcl} -\Delta u & = & \lambda \cdot \psi(u) \\ u & \in & \mathcal{H}_0(\Omega), \end{array} \right. \tag{2}$$

where Δ is the well-known Laplace operator and $\lambda \in \mathbb{R}$ is a parameter.

Wherever they appear, the vectors $\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$ and $\vec{\mathbf{k}}$ are the unit vectors of the axis Ox, Oy and Oz.

2.1 Optimization problem

It is well-known from Critical Point Theory (see [3][6]), that the (weak) solutions of equation (2) are precisely the critical points of the energy functional $\mathbb{I}_\lambda : \mathcal{H}_0(\Omega) \to \mathbb{R}$ attached to problem (2), where

$$\mathbb{I}_\lambda[u] = \frac{1}{2} \int_\Omega \|\nabla u(x)\|^2 dx - \lambda \int_\Omega \left(\int_0^{u(x)} \psi(t) dt \right) dx. \tag{3}$$

In particular, the local minima of functional $\mathbb{I}_\lambda$ are the solutions of differential equation (2).

Note, that due to condition (1), the energy functional $\mathbb{I}_\lambda$ is coercive, i.e.,

$$\mathbb{I}_\lambda[u] \to +\infty \text{ as } \|u\|_{\mathcal{H}_0(\Omega)} \to +\infty, \forall \lambda \in \mathbb{R}.$$

Moreover, the functional $\mathbb{I}_\lambda$ is bounded from below and satisfies the Palais-Smale compactness condition [6]. Thus, at least *one* weak solution is always expected as the *global* minimum of the functional $\mathbb{I}_\lambda$ on the space $\mathcal{H}_0(\Omega)$.

Our aim is to find not only the global minimum, but also other local minima of the abovementioned energy functional.

B. Ricceri [4] has predicted theoretically the existence of other local minimum points of the functional $\mathbb{I}_\lambda$ for certain parameters $\lambda \in \mathbb{R}$ (when the nonlinear ψ fulfills the condition (1)). According to this fact, we restrict our attention to study the following multimodal optimization problem:

$$\begin{cases} \mathbb{I}_\lambda[u] \to \text{minimize} \\ u \in \mathcal{H}_0(\Omega). \end{cases} \tag{4}$$

Proposed technique evolves candidate solutions in the search space $\mathcal{H}_0(\Omega)$. Our approach relies on bicubic B-spline approximation [1] for generating C^2-continuous Euler-Monge surfaces defined on domain Ω. The boundary curves of the generated surface lie in the plane *xOy*.

2.2 Bicubic B-spline patches as building stones of C^2-continuous approximated surfaces having two boundary curves

Let us denote by

$$M = [P_{ij}]_{i=0..3,\ j=0..3} \in \mathbb{M}_{4,4}(\mathbb{R}^3)$$

a controlnet, which determines the shape of a bicubic patch.

Consider the fourth ordered periodic blending functions (polynomials of degree three) of this approximation method, i.e.,

$$\begin{cases} f_0(t) &= & (1-t)^3/6 \\ f_1(t) &= & \left[3(1-t)^2 t + 3(1-t) + 1\right]/6 \\ f_2(t) &= & \left[3(1-t)t^2 + 3t + 1\right]/6 \\ f_3(t) &= & t^3/6. \end{cases}$$

The matrix representation of the bicubic B-spline patch $u : [0,1] \times [0,1] \to \mathbb{R}^3$ is given by

$$u(p,r) = \begin{bmatrix} f_0(p) & f_1(p) & f_2(p) & f_3(p) \end{bmatrix} \cdot M \cdot \begin{bmatrix} f_0(r) \\ f_1(r) \\ f_2(r) \\ f_3(r) \end{bmatrix}. \tag{5}$$

The unit normal vector of a bicubic B-spline patch $u : [0,1] \times [0,1] \to \mathbb{R}^3$ in the point $(p,r) \in [0,1] \times [0,1]$ is denoted by $\vec{n}_u(p,r)$.

Let Ω a non-empty and compact subset of $\mathbb{R}^2$, which has a hole inside. We suppose, that the two boundary curves of this domain are given in polar coordinates and they are denoted by

$$\gamma_i : [0,2\pi] \to \mathbb{R}^2,\ \gamma_i(\varphi) = (x_i(\varphi), y_i(\varphi)) = (\rho_i(\varphi) \cdot \cos\varphi, \rho_i(\varphi) \cdot \sin\varphi),\ i = 1,2.$$

We assume, that these planar curves respect the following conditions:

- $\gamma_1, \gamma_2 \in C^2([0,2\pi], \mathbb{R}^2)$, i.e., the boundary curves of the domain are smooth;
- $\rho_1(\varphi) > \rho_2(\varphi) > 0,\ \forall\, \varphi \in [0,2\pi]$, i.e., γ_1 is the outer boundary curve and γ_2 is the inner one;
- $\gamma_i(0) = \gamma_i(2\pi)$ and $\rho_i(\varphi_1) \neq \rho_i(\varphi_2),\ \forall \varphi_1, \varphi_2 \in]0, 2\pi[,\ i = 1,2$, i.e., the boundaries are closed curves without any self-intersections.

Consider the index sets $\mathcal{K} := \{0,1,...,m\}$ and $\mathcal{L} := \{1,...,n\}$, and the division points

$$t_i = \frac{i}{m-1},\ \varphi_j = j \cdot \frac{2\pi}{n},\ i \in \{0,1,...,m-1\},\ j \in \{0,1,...,n-1\}.$$

For all indices $i \in \{0,1,...,m-1\},\ j \in \{0,1,...,n-1\}$, let z_{ij} a uniformly distributed random number from the range $[a,b]$ defined as

$$z_{ij} = \begin{cases} 0, & \text{if } i = 0 \text{ or } i = m-1 \\ a + (b-a) \cdot \text{unifrnd}[0,1], & \text{if } i \in \{1,2,..,m-2\}. \end{cases}$$

Consider the vector matrix $P = [P_{ij}]_{i=0,1,...m-1,\ j=0,1,...,n-1} \in \mathbb{M}_{m,n}(\mathbb{R}^3),\ m \geq 3,\ n \geq 3,$

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \cdots & P_{0,n-1} \\ P_{10} & P_{11} & P_{12} & \cdots & P_{1,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{m-2,0} & P_{m-2,1} & P_{m-2,2} & \cdots & P_{m-2,n-1} \\ P_{m-1,0} & P_{m-1,1} & P_{m-1,2} & \cdots & P_{m-1,n-1} \end{bmatrix}, \tag{6}$$

where

$$P_{ij} = ((1-t_i) \cdot x_1(\varphi_j) + t_i \cdot x_2(\varphi_j), (1-t_i) \cdot y_1(\varphi_j) + t_i \cdot y_2(\varphi_j), z_{ij}).$$

By attaching to the vector matrix P three new columns to the right, which are the duplications of the first three columns of the matrix P, and repeating the first and last rows in this new vector matrix, we get a controlnet denoted by $Q = [Q_{ij}]_{i=0..m+1,\, j=0..n+2} \in \mathbb{M}_{m+2,n+3}(\mathbb{R}^3)$,

$$Q = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \cdots & P_{0,n-1} & P_{00} & P_{01} & P_{02} \\ P_{00} & P_{01} & P_{02} & \cdots & P_{0,n-1} & P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} & \cdots & P_{1,n-1} & P_{10} & P_{11} & P_{12} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ P_{m-2,0} & P_{m-2,1} & P_{m-2,2} & \cdots & P_{m-2,n-1} & P_{m-2,0} & P_{m-2,1} & P_{m-2,2} \\ P_{m-1,0} & P_{m-1,1} & P_{m-1,2} & \cdots & P_{m-1,n-1} & P_{m-1,0} & P_{m-1,1} & P_{m-1,2} \\ P_{m-1,0} & P_{m-1,1} & P_{m-1,2} & \cdots & P_{m-1,n-1} & P_{m-1,0} & P_{m-1,1} & P_{m-1,2} \end{bmatrix}. \tag{7}$$

For an arbitrary row $k \in \mathcal{K}$ and column index $\ell \in \mathcal{L}$ of the vector matrix Q, let us introduce the functions

$$r_k, c_\ell : \{-1,0,1,2\} \to \mathbb{N},$$

defined as

$$r_k(i) = \begin{cases} 0, & \text{if } k+i = -1 \\ m+1, & \text{if } k+i = m+2 \\ k+i, & \text{in all other cases,} \end{cases}$$

$$c_\ell(j) = \ell + j.$$

For all index-pairs $(k,\ell) \in \mathcal{K} \times \mathcal{L}$, consider the vector matrices $M_{k,\ell} \in \mathbb{M}_{4,4}(\mathbb{R}^3)$ defined by

$$M_{k,\ell} = [Q_{r_k(i),c_\ell(j)}]_{i=-1,0,1,2,\ j=-1,0,1,2.} \tag{8}$$

Applying formula (5), we can generate the shape of the bicubic B-spline patches $u_{k,\ell} : [0,1] \times [0,1] \to \mathbb{R}^3$ determined by the vector matrices $M_{k,\ell},\ \forall (k,\ell) \in \mathcal{K} \times \mathcal{L}$.

Chosing the elements of the vector matrices $M_{k,\ell}, \forall (k,l) \in \mathcal{K} \times \mathcal{L}$, as described above, we can ensure, that the adjacent patches are connecting to each other with C^2-continuity, the surface's shape generated by all patches is closed in the direction of φ, and the two boundaries of the surface lie in the plane xOy.

Consider the compact subsets $\Omega_{k,\ell} \subset \mathbb{R}^2,\ \forall\ (k,\ell) \in \mathcal{K} \times \mathcal{L}$, which are defined as

$$\Omega_{k,l} = \left\{ \left(u_{k,\ell}(p,r) \cdot \vec{\mathbf{i}}, u_{k,\ell}(p,r) \cdot \vec{\mathbf{j}} \right) \mid (p,r) \in [0,1] \times [0,1] \right\}.$$

Increasing the row or/and column dimension of the vector matrix (6), the union $\bigcup_{(k,\ell) \in \mathcal{K} \times \mathcal{L}} \Omega_{k,\ell}$ approximates the domain Ω. With the described method, we can generate randomly C^2-continuous Euler-Monge surfaces, which lie over the approximated domain Ω.

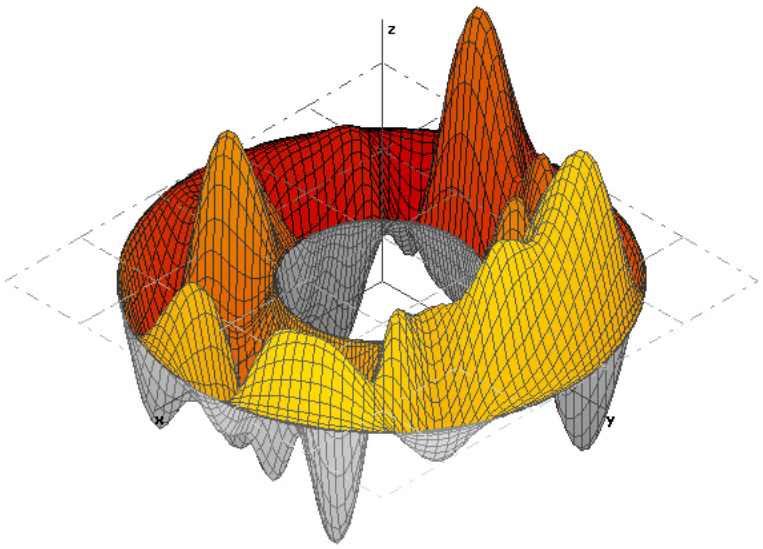


Figure 1: Randomly generated C^2-continuous Euler-Monge type B-spline surface over a compact domain of $\mathbb{R}^2$ with one hole inside. The dimension of the used vector matrix (6) is 5×30.

Figure 1 depictes a randomly generated C^2-smooth Euler-Monge type B-spline surface over the approximated domain

$$\Omega = \left\{ (\rho \cdot \cos\varphi, \rho \cdot \sin\varphi) \in \mathbb{R}^2 | (\rho, \varphi) \in [2,5] \times [0, 2\pi] \right\}.$$

A functional, which approximates the value of the energy functional (3), has the form:

$$\mathbb{J}_{\lambda}^{\circledcirc}[u] = \sum_{k \in \mathcal{K}} \sum_{\ell \in \mathcal{L}} \int_0^1 \int_0^1 \left\{ \left[\frac{1}{2} \cdot \| \nabla (u_{k,\ell}(p,r) \cdot \vec{\mathbf{k}}) \|^2 - \lambda \cdot \left(\int_0^{u_{k,\ell}(p,r) \cdot \vec{\mathbf{k}}} \psi(t)\, dt \right) \right] \cdot \left| J_{k,\ell}(p,r) \right| \right\} dp\, dr, \qquad (9)$$

where u is an Euler-Monge type B-spline surface having a shape consisting of all patches $u_{k,\ell}$ determined by the vector matrices (8), and

$$J_{k,\ell}(p,r) = \det \begin{bmatrix} \frac{\partial u_{k,\ell}}{\partial p}(p,r) \cdot \vec{\mathbf{i}} & \frac{\partial u_{k,\ell}}{\partial r}(p,r) \cdot \vec{\mathbf{i}} \\ \frac{\partial u_{k,\ell}}{\partial p}(p,r) \cdot \vec{\mathbf{j}} & \frac{\partial u_{k,\ell}}{\partial r}(p,r) \cdot \vec{\mathbf{j}} \end{bmatrix}, \ (p,r) \in [0,1] \times [0,1], \ (k,\ell) \in \mathcal{K} \times \mathcal{L}.$$

3 Evolutionary approach

A slightly modificated version of the evolutionary roaming algorithm presented in [5] is proposed for solving the multimodal optimization problem (6). Roaming is an evolutionary technique for detecting all optima of the functions of type

$$f : [a_1, b_1] \times \cdots \times [a_n, b_n] \to \mathbb{R}, n \geq 1.$$

This section specifies how the elements and operators of the roaming algorithm from [5] are instantiated for detecting all the minima of a coercive energy functional.

Roaming technique is based on the Euclidean distance of the individuals (i.e., n-dimensional vectors) involved in the search process. Our approach uses a different distance concept.

3.1 Evolutionary representation

Initial *individuals* of the *(sub)population(s)* involved in the search process(es), are randomly genarated C^2-continuous Euler-Monge type B-splines surfaces over the given and approximated compact domain Ω.

Each individual is determined by a controlnet of type (7), which is based on some vector matrix of type (6). The row and column dimensions of the controlnets are the same for all individuals and they do not change during the evolutionary process.

A controlnet is determined by randomly generated 3-dimensional controlpoints from the approximated domain $\Omega \times [a,b] \subset \mathbb{R}^3$ (as described in the previous section). So, the controlpoints correspond to the *genes*.

In each individual's controlnet the controlpoints in the first and last two rows are fixed (i.e., they are not involved in any mutational operations), because they determine the two fixed boundary curves.

An individual determined by a controlnet

$$Q = [Q_{ij}]_{i=0..m+1,\ j=0..n+2}$$

of type (7) based on the vector matrix

$$P = [P_{ij}]_{i=0..m-1,\ j=0..n-1} \in \mathbb{M}_{m,n}(\mathbb{R}^3),$$

is denoted by

$$u^{\circledcirc}(P_{ij}(x_i,y_j,z_{ij}))_{i=0..m-1,\ j=0..n-1},$$

or simply by $u^{\circledcirc}$. This individual consists of the bicubic B-spline pacthes

$$u_{k,\ell} : [0,1] \times [0,1] \to \mathbb{R}^3,\ (k,\ell) \in \mathcal{K} \times \mathcal{L}$$

determined by the vector matrices (8).

A search process starts with a randomly generated initial subpopulation

$$SP_0^{\circledcirc} = \{u_1^{\circledcirc}, u_2^{\circledcirc}, ..., u_N^{\circledcirc}\}$$

containing the individuals

$$u_\alpha^{\circledcirc}(P_{ij}^\alpha(x_i,y_j,z_{ij}^\alpha))_{i=0..m-1,\ j=0..n-1},\ \alpha \in \{1,2,...,N\}$$

satisfying the given boundary constraints. As the subpopulation evolves from the actual generation $SP_g^{\circledcirc}$ to the next generation $SP_{g+1}^{\circledcirc}$, $g \geq 0$, the number of individuals remains the same.

The *fitness* of an individual $u_\alpha^{\circledcirc} \in SP_g^{\circledcirc},\ \alpha \in \{1,2,...,N\},\ g \geq 0$, is defined as

$$eval(u_\alpha^{\circledcirc}) = -\mathbb{J}_\lambda^{\circledcirc}[u_\alpha^{\circledcirc}].$$

Consider the individuals

$$u_\alpha^{\circledcirc}(P_{ij}^\alpha(x_i,y_j,z_{ij}^\alpha))_{i=0..m-1,\ j=0..n-1}$$

and

$$u_\beta^{\circledcirc}(P_{ij}^\beta(x_i,y_j,z_{ij}^\beta))_{i=0..m-1,\ j=0..n-1},$$

which consist of the patches

$$u_{k,\ell}^\alpha,\ u_{k,\ell}^\beta : [0,1] \times [0,1] \to \mathbb{R}^3,\ (k,\ell) \in \mathcal{K} \times \mathcal{L}.$$

Let a fine division grid

$$\Delta_\square = \left\{ \left(\frac{i}{m_p}, \frac{j}{n_r} \right) \right\}_{i=0..m_p, j=0..n_r}$$

of the domain $[0,1] \times [0,1]$, and let us consider the fixed numbers $\theta \in \left]0, \frac{\pi}{2}\right]$ and $\varepsilon > 0$. Finally, for all $(k,\ell) \in \mathcal{K} \times \mathcal{L}$ let us construct the following sets:

$$N_{k,\ell}^\theta = \left\{ (p,r) \in \Delta_\square \,\middle|\, \vec{n}_{u_{k,\ell}^\alpha}(p,r) \cdot \vec{n}_{u_{k,\ell}^\beta}(p,r) \leq \cos\theta \right\},$$

$$D_{k,\ell}^\varepsilon = \left\{ (p,r) \in \Delta_\square \,\middle|\, \|u_{k,\ell}^\alpha(p,r) - u_{k,\ell}^\beta(p,r)\| > \varepsilon \right\}.$$

The *distance* between the individuals $u_\alpha^{\circledcirc}$ and $u_\beta^{\circledcirc}$ is defined by

$$d_\varepsilon^\theta(u_\alpha^{\circledcirc}, u_\beta^{\circledcirc}) = \max_{(k,\ell) \in \mathcal{K} \times \mathcal{L}} \frac{\mathrm{card} N_{k,\ell}^\theta + \mathrm{card} D_{k,\ell}^\varepsilon}{2 \cdot (m_p+1) \cdot (n_r+1)}.$$

3.2 Search operators

The *selection*, *recombination* and the *mutation* operators are presented in the following part of this section.

- **Selection** Tournament selection was used for determining the individuals of the next generations.
- **Recombination** Let us consider a randomly coupled pair of individuals from the actual generation of a subpopulation. Recombination is based on the following mating process: first we select randomly a point in the inner region of the parents' vector matrices of type (6), then the complementary parts of the parents' vectors in blocks determined by the selected point are used to form the genes of the offsprings.

  Figure 2 illustrates eight recombination operators: switching the controlpoints corresponding to the dark and light grayed parts of the parents' vector matrices, we get two new vector matrices of type (6), which are used to build two new controlnets of type (7). So, we can generate two offsprings.

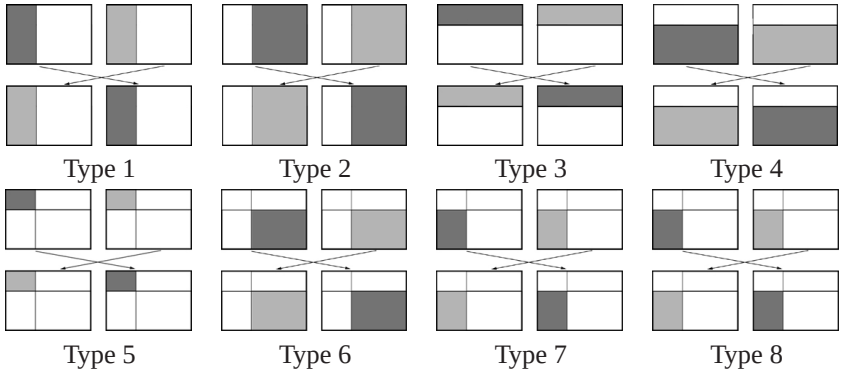


Figure 2: Several recombination operators

  The resulting offsprings, the parents and the worst individual of the actual subpopulation compete for survival.

- **Mutation** Consider the randomly selected individual $u^{\circledcirc}(P_{ij}(x_i,y_j,z_{ij}))_{i=0..m-1,j=0..n-1}$ from the actual generation of a subpopulation.

  Consider the index sets $I=\{1,2,...,m-2\}$, $J=\{0,1,...,n-1\}$, and let us suppose, the vector

$$P_{ij}(x_i,y_j,z_{ij}),\ (i,j)\in I\times J$$

  has been selected randomly for mutation.

  Mutation is done by perturbing the last coordinate $z_{ij}\in[a,b]$ of the selected controlpoint P_{ij} with the function

$$\mathrm{mut}(z_{ij})=\zeta_s(\lfloor \mathrm{unifrnd}(0,1)\cdot 5\rfloor),$$

  where

$$\zeta_s:\{0,1,2,3,4\}\to[a,b], s\geq 2, \text{ and } \begin{cases} \zeta_s(0) &= a+\mathrm{unifrnd}(0,1)\cdot(b-a)\\ \zeta_s(1) &= \max\left(z_{ij}-\mathrm{unifrnd}(0,1)\cdot\frac{|b-a|}{s},a\right)\\ \zeta_s(2) &= \min\left(z_{ij}+\mathrm{unifrnd}(0,1)\cdot\frac{|b-a|}{s},b\right)\\ \zeta_s(3) &= a\\ \zeta_s(4) &= b. \end{cases}$$

  The domain $\Omega\times[a,b]\subset\mathbb{R}^3$ has been splitted into $s\geq 2$ subdomains by paralel planes having normal $\vec{\mathbf{k}}$. Increasing the value of s, we can ensure fine shape modifications (perturbations) of the individuals.

  The offspring and its parent compete for survival. In the process of redirecting subpopulations towards other search regions of the search space $\mathcal{H}_0(\Omega)$, offsprings always replace their parents.

3.3 Roaming optimization

The roaming technique, propesed in [5], identifies the local/global optima using isolated subpopulations and stores them in an external population called archive.

Let us suppose that our initial population consists of M subpopulation

$$\mathbb{P}_0 = \{SP_0^1, SP_0^2, \cdots, SP_0^M\},$$

and let us assume that each subpopulation has N individuals. A stability measure for subpopulations is defined and by using it, subpopulations are evaluated as σ-stable or σ-unstable, where $\sigma \in [0,1]$.

The stability measure of a g-generational subpopulation $SP_g^k,\ g \geq 0,\ k \in \{1,2,...,M\}$ is defined as

$$SM(SP_g^k) = 1 - \frac{\text{card}\{u \in SP_{g+1}^k |\ eval(u) > eval(u^\star)\}}{N},$$

where $u^\star$ is the best individual of generation SP_g^k.

The subpopulation $SP_g^k,\ k \in \{1,2,...,M\}$ is called σ-stable (σ-unstable), if the condition

$$SM(SP_g^k) \geq \sigma$$

holds (it is not fulfilled).

1-unstable subpopulations evolve in isolation until they become 1-stable. The best individual in a 1-stable subpopulation is considered as a potential local optimum, which is saved into the archive with the help of a parameter $\delta_\varepsilon^\theta > 0$. The number $\delta_\varepsilon^\theta$ depends from the preliminary fixed numbers $\theta \in]0, \frac{\pi}{2}]$ and $\varepsilon > 0$, and it is related to the minimum distance between two optima. Before adding a candidate optimum $u^\star$ to the archive, the distance between $u^\star$ and every already archived optimum a is compared with this parameter. If the condition $d(u^\star, a) \leq \delta_\varepsilon^\theta$ holds, then only the best fitted individuals from $u^\star$ and a remains in the archive. If the condition $d(u^\star, a) > \delta_\varepsilon^\theta$ holds for all other already archived individual a, then $u^\star$ represents a new optimum for the archive.

After adding a potential optimum to the archive, to avoid the search process to get stuck, the search performed by all the τ-stable ($\tau \in]1-\eta, 1], \eta \to 0_+$; a treshold parameter) subpopulations must be redirected (i.e., a strong non-conditional mutation must be applied) towards other regions of the search space $\mathcal{H}_0(\Omega)$.

4 Example

Due to the limited pagenumber, just a single numerical experiment is described. Consider the compact domain $\Omega \subset \mathbb{R}^2$,

$$\Omega = \left\{(\rho \cdot \cos\varphi, \rho \cdot \sin\varphi) \in \mathbb{R}^2 | (\rho, \varphi) \in [2,5] \times [0, 2\pi]\right\}.$$

Let the function $\psi : \mathbb{R} \to \mathbb{R}$,

$$\psi(t) = \sqrt{|t|}\arctan t,$$

which fulfills the growth condition (1) with the parameters $s = \frac{1}{2}$ and $c = \frac{\pi}{2}$.

Consider the energy functional

$$\mathbb{I}_{-2}[u] = \frac{1}{2}\int_\Omega \|\nabla u(x)\|^2\, dx + 2\int_\Omega \left(\int_0^{u(x)} \psi(t)dt\right) dx, \tag{10}$$

and let us approximate the solutions of the optimization problem

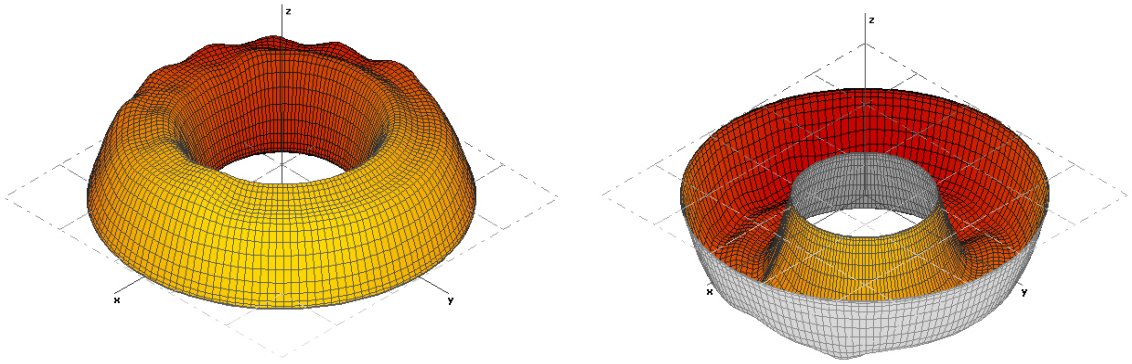
$$\begin{cases} \mathbb{I}_{-2}[u] \to \text{minimize} \\ u \in \mathcal{H}_0(\Omega). \end{cases}$$

The parameters of the proposed evolutionary roaming technique are:

- number of subpopulations: $M = 10$;
- number of individuals in each subpopulations: $N = 20$;
- the minimum distance between two optima: $\delta_{0.8}^{\frac{\pi}{6}} = 0.65$;

- the treshhold parameter for roaming subpopulations: $\tau = 0.9$;
- the probability of mutation for each subpopulation: $p_{mut} = 0.8$;
- the probability of recombination for each subpopulation: $p_{rec} = 0.6$;
- the dimension of the vector matrices, which generate the individuals: $m \times n = 5 \times 30$;
- the last coordinate of the controlpoints are generated from the range: $[a,b] = [-10,10]$.

After 2500 generations of the proposed algorithm, the archive contains two C^2-continuous Euler-Monge type B-spline surfaces. Figure 3 depictes the shape of these surfaces. The value of the energy functional (10) in the case of surface **3a**(**3b**) is -121.831130981445(-121.275131225586).



3a **3b**

Figure 3: Two local minima of the energy functional (10)

Naturally, increasing the dimension of the used vector matrices, someone could approximate these local minima with less error, but the algorithm will be very time-consuming. Implementing the proposed algorithm in a local area network conform a "master and slaves" model (each slave may represent a subpopulation and the master may be the archive), the needed time to find better solutions can be reduced.

5 Summary. Conclusions and further work

Standard literature does not describe an exact or numerical method determining the local minima of the coercive energy functionals.

In most cases there are only existence theorems for critical points (in particular, for local minima) which are weak solutions of equations of type (2).

In standard literature, the disc is the most frequent, non-empty and compact domain of $\mathbb{R}^2$ on which the partial differential equation (2) is defined. We proposed a method to construct randomly C^2-continuous Euler-Monge type B-spline surfaces, which lie over a given compact domain having one hole inside.

Note, that the function $\psi : \mathbb{R} \to \mathbb{R}$ appeared in problem (2) is not necessarily continuous, thus, the energy functional (3) may not be differentiable, only a locally Lipschitz function. Hence, the optimization problem (6) may not be solved by other classical and well-known numerical methods (e.g., Newton method).

The paper proposes an evolutionary roaming algorithm to detect and construct local minima for coercive energy functionals. Proposed evolutionary technique will be tested for other types of compact domains. The algorithm will be extended for dealing with strongly constrained problems, too.

Acknowledgement

This work has been supported by grants from the Research Programs Institute of Foundation Sapientia, Cluj Napoca, Romania. The authors also thank Alexandru Kristály from Babeş-Bolyai University, Cluj Napoca, Romania, for his encouragements, suggestions and remarks.

References

[1] J. Hoschek, D. Lasser: *Fundamentals of Computer Aided Geometric Design*, A K Peters Ltd., Wellesley, Massachusetts, USA, 1993.

[2] P. D. Panagiotopoulos: *Hemivariational Inequalities: Applications to Mechanics and Engeneering*, Springer-Verlag, New-York, 1993.

[3] D. Motreanu, P. D. Panagiotopoulos: *Minimax theorems and qualitative properties of the solutions of hemivariational inequalities*, Kluwer Academic Publishers, Dordrecht, 1999.

[4] B. Ricceri: *Existence of three solutions for a class of elliptic eigenvalue problems. Nonlinear operator theory.* Math. Comput. Modelling 32, no. 11-13, 1485–1494, 2000.

[5] R. I. Lung, D. Dumitrescu: *Roaming Optimization: A New Evolutionary Technique for Multimodal Optimization*, Studia Univer. Babeş-Bolyai, Informatica, Volume XLIX, Number 1, 2004.

[6] A. Kristály, Cs. Varga: *An Introduction to Critical Point Theory for Non-Smooth Functions*, Stientia Publishing House, Cluj-Napoca, 2004.

Dan Dumitrescu, Ágoston Róth
Babeş-Bolyai University
Department of Computer Science
Address: 400084 Cluj-Napoca, Romania
E-mail: ddumitr@cs.ubbcluj.ro, agoston_roth@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 41-51

Possibility and Probability in Fuzzy Theory

Invited paper

Angel Garrido

Abstract: In the problems of Fuzzy Theory, it is very important to introduce the notion of distance between Fuzzy Sets, or Fuzzy Distance. Because from this, we can obtain the measure of fuzziness, or degree of how fuzzy a set is. This opens the perspective to analyze the concepts of *possibility* and *probability*, and so on.

Keywords: Mathematical Analysis, Probability Theory, Fuzzy Inference, A. I.

1 Introduction

The origins of the Possibility Theory are in Zadeh. But many efforts, from then, have improved more and more this field. For instance, the papers and books of Dubois and Prade.

Our purpose is to give, firstly, a panoramic vision of the known results, expanding their boundaries with some new ideas. For instance, the measure of fuzziness through the Shannon's concept of entropy or by the different metric distances.

2 Difference and Distance in Fuzzy Sets

If we take two sets, A and B, the *difference* is given by:

$$A - B = A \cap c(B)$$

There exist two ways of obtaining the difference between fuzzy sets:
By *simple method*:
For instance, if we take:

$$A = \{a \mid 0.1,\ b \mid 0.3,\ c \mid 0.6,\ d \mid 0.9\}$$
$$B = \{a \mid 0.2,\ b \mid 0.5,\ c \mid 0.8,\ d \mid 1\}$$

then:

$$c(B) = \{a \mid 0.8,\ b \mid 0.5,\ c \mid 0.2,\ d \mid 0\}$$

therefore:

$$A - B = A \cap c(B) = \{a \mid \min(0.1,\ 0.8),\ b \mid \min(0.3,\ 0.5), c \mid \min(0.6,\ 0.2),\ d \mid \min(0.9,\ 0)\} =$$

$$= \{a \mid 0.1, b \mid 0.3, c \mid 0.2, d \mid 0\}$$

While the *Bounded Difference* is defined through a new operator, θ, according to the membership function:

$$\mu_{A\,\theta\,B}(x) = \max\{\mu_A(x) - \mu_B(x),\ 0\}$$

So, in the example above:

$$A\ \theta\ B =$$
$$= \{a \mid \max\{\mu_A(a) - \mu_B(a),\ 0\}, b \mid \max\{\mu_A(b) - \mu_B(b),\ 0\}, c \mid \max\{\mu_A(c) - \mu_B(c),\ 0\},$$
$$d \mid \max\{\mu_A(d) - \mu_B(d),\ 0\}\} = \{a \mid 0,\ b \mid 0,\ c \mid 0,\ d \mid 0\}$$

Therefore, the elements a, b, c, d do not actually belong to $A\theta B$.

To introduce the *distance between fuzzy sets*, A and B, we consider different *possibilities,* now based on the values of the membership functions in the point $x \in U$:

i) the well known *Euclidean distance:*

$$e(A,\ B) = \left[\sum \{\mu_A(x) - \mu_B(x)\}^2\right]^{1/2}$$

ii) the *Hamming distance:*

$$d(A,\ B) = \sum |\mu_A(x_i) - \mu_B(x_i)|$$

with $i \in \{1,2,...n\}$ and $x_i \in U$, universe of discourse.

We can prove easily that:

$$1)\ d(A,B) \geq 0$$

$$2)\ d(A,B) = d(B,A)$$

$$3)\ d(A,C) \leq d(A,B) + d(B,C)$$

$$4)\ d(A,A) = 0$$

And also it can be defined the *relative Hamming distance* (δ), when the universal set U is finite, for instance, with n elements:

$$if\ \ \sharp(U) = n \Rightarrow \delta(A,B) = \tfrac{1}{n}\, d(A,B)$$

For instance, let A and B be as in the aforementioned example.
Then:

$$e(A,B) = \left[\{0.1-0.2\}^2 + \{0.3-0.5\}^2 + \{0.6-0.8\}^2 + \{0.9-1\}^2\right]^{1/2} = 0.316$$

$$d(A,B) = |0.1-0.2| + |0.3-0.5| + |0.6-0.8| + |0.9-1| = 0.6$$

$$\delta(A,B) = \tfrac{1}{n} d(A,B)$$

So, if $n = 4$, then:

$$\delta(A,B) = \tfrac{1}{4} d(A,B) = 0.15$$

And generalizing, we can also define the *Minkowskian distance:*

$$d_w(A,B) = [\sum |\mu_A(x) - \mu_B(x)|^w]^{1/w}$$

where $w \in [1,+\infty]$.

Observe that when $m = 1$, we obtain the Hamming distance.

And when $m = 2$, we find the Euclidean distance.

Both are particular cases, therefore, of Minkowskian distance.

3 Fuzzy Distance between Fuzzy Sets

We need to introduce the *Extension Principle*, according to which:

Starting with a Cartesian product of universal sets:

$$U = \prod U_i,\ i = 1,\ 2,\ ...,\ r$$

And a collection of fuzzy sets, each one into the corresponding universal set:

$$A_i \subseteq U_i,\ i = 1,\ 2,\ ...,\ r.$$

Then, we define the *Cartesian product of fuzzy sets:*

$$\prod A_i,\ i = 1,\ 2,\ ...,\ r$$

through their membership function:

$$\mu_{\prod A_i}(x_1,\ x_2,\ ...,\ x_r) \equiv \min\left\{\mu_{A_1}(x_1),\ \mu_{A_2}(x_2),\ ...,\ \mu_{A_r}(x_r)\right\}$$

Let F be the function from the universe U to the universe V.

Then, the fuzzy set:

$$B \subseteq V$$

can be obtained by F and the collection of fuzzy sets, $\{A_i\}_{i=1}^{r}$, in this way:

$$\mu_B(y) = 0,\ \text{if}\ F^{-1}(y) = \emptyset$$

and

$$\mu_B(y) = \max_{y=F(x_1,x_2,...,x_r)} \left[\min\left\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), ..., \mu_{A_r}(x_r)\right\}\right],\ if\ F^{-1}(y) \neq \emptyset$$

If the function F is *one-to-one*, we have:

$$\mu_B(y) = \mu_A\left(F^{-1}[y]\right),\ \text{when}\ F^{-1}(y) \neq \emptyset$$

Let $(U,\ d)$ be a *pseudometric space*. Therefore, with:

$$d : U^2 \to R_+ \cup \{0\}$$

such that verifies:

$$1)\ d(x,x)=0,\ \forall x\in U$$

$$2)\ d(x,y)=d(y,x),\forall x,y\in U$$

$$3)\ d(x,z)\leq d(x,y)+d(y,z),\forall x,y,z\in U$$

Remember also that the additional condition:

$$4)\ if\ d(x,y)=0,\ then\ x=y$$

transforms d into a distance, and in such case,$(U,\ d)$ is a *metric space*.

In our pseudometric space, $(U,\ d)$, if we take two fuzzy subsets, A and B, it is possible to introduce by the extension principle the *pseudometric distance* (also a fuzzy set) *between A and B* :

$$\forall\rho\in R_{+},\ \mu_{d(A,B)}(\rho)=\max_{\rho=d(a,b)}\left[\min\left\{\mu_{A}(a),\mu_{B}(b)\right\}\right]$$

4 Probability and Possibility

A very frequent question turns about the relation between Probability Theory and Fuzzy Theory. More concretely, if there exists some contradiction between both. It is a very natural question, because both the values of the Probability Distribution, P, and of the Membership Function, μ, belong to the closed unit interval. For both, the range is $[0,\ 1]$. And this suggest, at least, some degree of kinship.

We suppose well known the axiomatic of Kolmogorov for the probability and its corollaries, and now introduce the fuzzy notion of *possibility:*

Suppose the fuzzy set A on the universal set U. Such fuzzy set is defined by the assignation of values, in the range $[0,\ 1]$, through the membership function, μ. So, the $\mu_{A}(x)$ represent the value on x of the possibility distribution function for A in U. In this way, adjoining the values $\mu_{A}(x)$, for each x, we have defined the fuzzy set.

As you know, the sum of probabilities on all the events, in the sample space, must be equal to 1. But this is not so necessarily, in the case of possibilities on fuzzy sets, into the universal set U.

For instance, we can have:

x	1	2	3
$\mu(x)$	0.8	0.9	1
$P(x)$	0.5	0.3	0.1

Firstly, we can observe the non identical behaviour between probabilities and possibilities. Higher possibilities not always corresponds to higher probabilities.

Given an event, A, *the possibility acts as upper bound of the probability:*

$$P(A)\leq\mu(A)$$

If we suppose $\{A_i\}_{i=1}^{n}$, *mutually exclusive fuzzy sets*, we have:

$$\mu(\cup A_i)=\max\{\mu(A_i)\}$$

and if we take $\{A_i\}_{i=1}^{n}$ as *independent* fuzzy sets:

$$\mu\left(\cap A_i\right) = \min\left\{\mu\left(A_i\right)\right\}$$

In the classical Probability theory the boundary of an event is always precise. But it is not the case when we work with fuzzy situations.

When we deal with the probability of fuzzy events, there are two ways: the probability as a crisp set or either as a fuzzy set. In this case, we say *Fuzzy Probability.* In the last situation, for the Fuzzy Probability of a Fuzzy Event, we consider the fuzzy event:

$$A = \{x \mid \mu_A(x) : x \in S\}$$

where S is the sample space.

>From this, we take the usual $\alpha - cut$ of such event:

$$A_\alpha = \{x \mid \mu_A(x) : \mu_A(x) \geq \alpha\}$$

Then, the *probability of the* $\alpha - cut$ should be:

$$P\left(A_\alpha\right) = \sum_{x \in A_\alpha} P\left(x\right)$$

being A_α the union of mutually exclusive events.

Therefore, the *probability of* A_α is the sum of the probability of each event in the α-cut set. So, we can say that:

"the possibility of the probability of the set A_α *to be* $P\left(A_\alpha\right)$ *is precisely such* α*"*

Starting from this concept, we can introduce already the *Fuzzy Probability of Fuzzy Event A* as:

$$P\left(A\right) = \{P\left(A_\alpha\right) \mid \alpha : \alpha \in [0,\ 1]\}$$

For instance, if we start from: $S = \{a,\ b,\ c\}$, as sample space, being:

$$P\left(a\right) = 0.2$$

$$P\left(b\right) = 0.3$$

$$P\left(c\right) = 0.5$$

and we have the fuzzy event:

$$A = \{a \mid 1,\ b \mid 0.9,\ c \mid 0.8\}$$

this produces the $\alpha - cut\ sets$ (which are crisp sets):

$$A_\alpha = A,\ if\ \alpha < 0.8$$

$$A_{0.8} = \{a,\ b,\ c\} = A$$

$$A_{0.9} = \{a,\ b\},\ A_1 = \{a\}$$

As crisp sets, we can calculate now the probability of each $\alpha - cut$ event:

$$P(A_{0.8}) = 0.2 + 0.3 + 0.5 = 1$$

$$P(A_{0.9}) = 0.2 + 0.3 = 0.5$$

$$P(A_1) = 0.2$$

So, we can reach:

$$P(A) = \{1 \mid 0.8,\ 0.5 \mid 0.9,\ 0.2 \mid 1\}.$$

5 Uncertainty level and Measure of Fuzziness

If we take a set of elements with degrees of membership between 0 and 1, when would the uncertainty be the greatest?

For instance, if the possibilities of obtaining a job, for a group, A, of candidates, are: 0.1, 0.2, 0.4, 0.5, 0.7, 0.9, the maximum of uncertainty is reached in the fourth. While the first has almost no possibility to pass the proof, for the last the success is almost secure. Therefore, the uncertainty increases as we approach to 0.5, and decreases close to the extremes (0 and 1) for the membership degree values.

Suppose another group, B, with respective possibilities: 0.4, 0.5, 0.54, 0.6, 0.63, 0.7. Where is the uncertainty greater, in A or in B? It is clear: in B, because their values are closer to 0.5, and for this the uncertainty respect to A increases.

In the case C, when all the candidates show values 0.5, this is still more uncertain (the maximum uncertainty).

In such situations, we attempt to calibrate the degree of uncertainty, or equivalently, the *Measure of Fuzziness.*

We define the function:

$$f : P(U) \to R$$

where $P(U)$ is the power set of U, containing all the subsets of the universal set and R is the real line.

In such function we need to impose *three conditions or axioms:*

1) $f(A) = 0 \Longleftrightarrow A$ is a *crisp set*

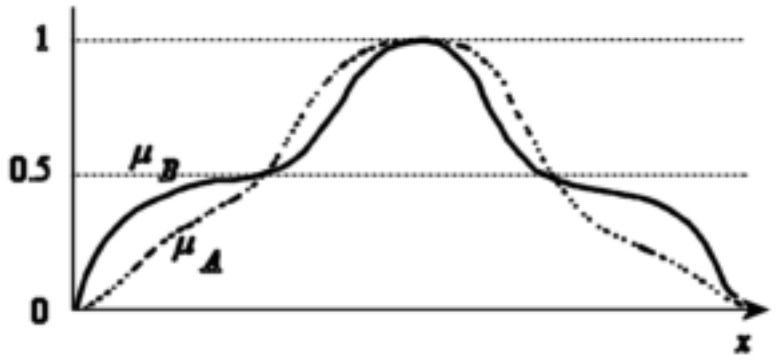
2) $A < B \Longleftrightarrow f(A) \leq f(B)$

That is, when the uncertainty of A is smaller than B, then the measured value $f(A)$ must be at most $f(B)$.

When we have $A < B$, we say that *A is sharper than B.*This signifies:

If we have two fuzzy sets, A and B, such that: $A < B$, the graphical situation must be as:

Therefore:



$$\mu_A(x) \leq \mu_B(x), if\ \mu(x) \leq \tfrac{1}{2}$$

and

$$\mu_A(x) \geq \mu_B(x), if\ \mu(x) \geq \tfrac{1}{2}.$$

The aforementioned equivalence: $A < B \Longleftrightarrow f(A) \leq f(B)$, implies the *monotonicity property.*

3) When the fuzziness reaches (for instance in B) its greatest degree, also the measure of its fuzziness, $f(B)$, should have the maximum value.

There are different ways to obtain the *measure of fuzziness*. For instance, through the *Shannon's Entropy* (H), or using *Metric Distances*.

Remember that H is used, in Information Theory, for measuring the amount of uncertainty in the information. H is defined by:

$$H[P(x)] = -\sum_{x \in X} P(x) \cdot \log_2 \{P(x)\}$$

for each $x \in U$.

Obviously, $P(x)$ denotes the probability distribution for x in the universal set U.

>From this, we can introduce the *measure of fuzziness, f*, as:

$$f(A) = -\sum_{x \in U} \{\mu_A(x) \cdot \log_2 \mu_A(x) + (1 - \mu_A(x)) \cdot \log_2 [1 - \mu_A(x)])\}$$

Observe that we add, in this expression, the uncertainties of the fuzzy set A and the uncertainties of its complement, $\bar{A}$, through their corresponding membership functions, $\mu_A(x)$ and $1 - \mu_A(x)$.

Also can be introduced the normalized measure, $\tilde{f}(A)$, of the measure $f(A)$, in this way:

$$\tilde{f}(A) = \tfrac{f(A)}{\sharp(U)}$$

where $\sharp(U)$ is the cardinal of the universal set U. It is clear that:

$$\tilde{f}(A) \in [0,\ 1]$$

verifying the above conditions 1) and 2).

We show now some *examples*, using the Shannon's entropy to obtain the *measure of fuzziness:*

We consider the probability distribution, *P.*

Let $U = \{a,\ b,\ c\}$ be the universal set.
Suppose that P is given by:

$$P(a) = P(b) = \tfrac{1}{4}$$

$$P(c) = \tfrac{1}{2}$$

The *uncertainty* is measured through the *Shannon's entropy, H* :

$$H(P) = -\left[\tfrac{1}{4}\cdot\log_2\tfrac{1}{4} + \tfrac{1}{4}\cdot\log_2\tfrac{1}{4} + \tfrac{1}{2}\cdot\log_2\tfrac{1}{2}\right] = 1.5$$

And about the aforementioned measure of fuzziness, for the fuzzy set:

$$A = \{a \mid 1,\ b \mid 0.2,\ c \mid 0.5\} \subset U$$

such degree of fuzziness of A is:

$$f(A) = -\left[\{1\cdot\log_2 1 + 0\} + \{0.2\cdot\log_2 0.2 + 0.8\cdot\log_2 0.8\} + \{0.5\cdot\log_2 0.5 + 0.5\cdot\log_2 0.5\}\right]$$

we can obtain:

$$f(A) = -\left[\tfrac{4}{5}\cdot\log_2\tfrac{4}{5} + \tfrac{1}{5}\cdot\log_2\tfrac{1}{5} + \log_2\tfrac{1}{2}\right] \simeq 1.7$$

and so, the normalized measure will be:

$$\tilde{f}(A) = \tfrac{f(A)}{\sharp(U)} \simeq 0.57$$

Observe that it verifies:

$$0 \leq \tilde{f}(A) \leq 1$$

while:

$$f\ (A) \notin [0,\ 1]$$

without problem.

In comparison with this case, if we take another fuzzy set, B, all its elements with the same membership degree, 0.5:

$$\mu_B(a) = \mu_B(b) = \mu_B(c) = \tfrac{1}{2}$$

then the measure of their fuzziness will be:

$$f(B) = -\{6 \cdot 0.5 \cdot \log_2 0.5\} = 3 > f(A) \simeq 1.7$$

As consequence, the fuzziness in B is greater than in A, as predictable, because the values of μ on its elements give the maximum uncertainty: when it is 0.5, or at least near to such value.

Furthermore, we compute the normalized measure of fuzziness as:

$$\tilde{f}(B) = \tfrac{f(B)}{\sharp(U)} = 1 > \tilde{f}(A) = 0.57$$

also a logical result: the preservation of the ordering from f to $\tilde{f}$.

Another way to compute the measure of fuzziness is by the concept of metric distance. So, starting with the Hamming distance or the Euclidean distance.

Suppose a fuzzy set, A. Its corresponding crisp set, C, can be defined by:

$$\mu_C(x) = 0, if\ \mu_A(x) \leq \tfrac{1}{2}$$

and

$$\mu_C(x) = 1, if\ \mu_A(x) > \tfrac{1}{2}$$

>From this, we can compute the measure of fuzziness through the distance between both sets, *A and C* (the first, fuzzy, and second, crisp set).

If we chose the *Hamming distance*, then the *measure of fuzziness* can be defined as:

$$f(A) = \sum_{x \in U} |\mu_A(x) - \mu_C(x)|$$

Whereas, if we consider the *Euclidean distance*, can be:

$$f(A) = \left[\sum_{x \in U} \{\mu_A(x) - \mu_C(x)\}^2 \right]^{1/2}$$

And in the case of the Minkowski's distance, must be:

$$f_w(A) = \left[\sum_{x \in U} |\mu_A(x) - \mu_C(x)|^w \right]^{1/w}$$

Moving on to the aforementioned A and B sets, our new purpose is to obtain the measure of fuzziness through the Hamming distance.

As you remember:

$$A = \{a \mid 1,\ b \mid 0.2,\ c \mid 0.5\} \subset U$$

$$B = \{a \mid 0.5,\ b \mid 0.5,\ c \mid 0.5\} \subset U$$

and then:

$$f(A) = |1-1| + |0.2-0| + |0.5-0| = 0.7$$

$$f(B) = |0.5-0| \cdot 3 = 1.5$$

As you can see directly, it verifies:

$$f(A) < f(B)$$

It also can be normalized, as in the previous case.
In the second term into differences:

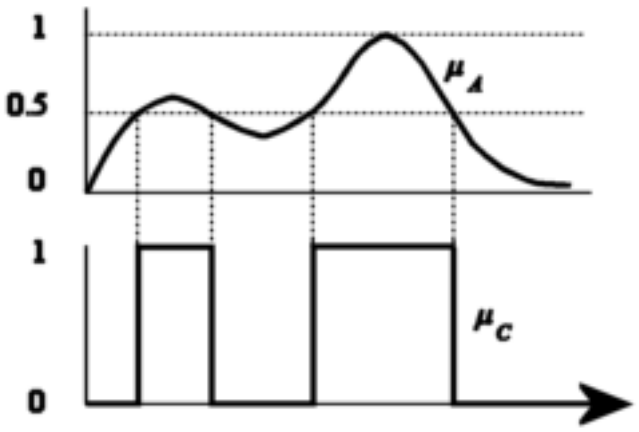
$$\sum_{x \in U} |\mu_A(x) - \mu_C(x)|$$

it appears the values of the membership function of the associated crisp set, C, that is, $\mu_C(x)$, reached through:

$$\mu_C(x) = 0,\ if\ \mu_A(x) \leq \tfrac{1}{2}$$

and

$$\mu_C(x) = 1,\ if\ \mu_A(x) > \tfrac{1}{2}$$



Therefore:

$$\mu_C(a) = 1$$

$$\mu_C(b) = \mu_C(c) = 0$$

in the first case, i.e., for the fuzzy set A.

Whereas in the second case, for the fuzzy set B:

$$\mu_C(a) = \mu_C(b) = \mu_C(c) = 0$$

If we consider the Euclidean distance, our *measure of fuzziness* must be:

$$f(A) = \left[\sum_{x\in U} \{\mu_A(x) - \mu_C(x)\}^2\right]^{1/2} = \sqrt[2]{(1-1)^2 + (0.2-0)^2 + (0.5-0)^2} = 0.5385$$

And in the final and more general case of Minkowskian distance:

$$f_w(A) = \left[\sum_{x\in U} |\mu_A(x) - \mu_C(x)|^w\right]^{1/w} = \sqrt[w]{(1-1)^w + (0.2-0)^w + (0.5-0)^w}$$

depending, obviously, on the values of w.

6 Final Note

With these considerations about uncertainty and fuzziness, their measure, through fuzzy techniques, we conclude our analysis.

References

[1] De Luca and Termini, "A Definition of a Non-Probabilistic Entropy in the Setting of Fuzzy and Sets Theory", *Information and Control*, 20 (1972) 301-312.

[2] Diamond and Kloeden, *Metric Spaces of Fuzzy Sets*, World Scientific, New Jersey, 1994.

[3] Dubois and Prade, *Possibility Theory. An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York, 1988.

[4] Pedrycz, *Fuzzy Control and Fuzzy Systems*, Wiley & Sons, New York, 1993.

[5] Wang and Klir, *Measure Fuzzy Theory.* Plenum Press, New York, 1997.

[6] Zadeh, "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems* 1 (1978) 3-28.

Angel Garrido
UNED, Faculty of Science
Departamento de Matemáticas Fundamentales
E-mail: algbmv@telefonica.net

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 53-54

A Computational Intelligence Approach to VRSDP (Vehicle Routing, Scheduling, and Dispatching Problems)

Invited paper

Kaoru Hirota, Fangyan Dong, Kewei Chen

Abstract: The objective of the Vehicle Routing, Scheduling, and Dispatching Problem (VRSDP) is to produce a delivery schedule for a group of vehicles, with respect to multiple users, so that while satisfying constraints delivery cost corresponding to users' orders is minimized. The VRSDP i.e., using a truck to deliver goods from a depot to supermarkets or retail stores - is widely observed in daily life. Solutions to such real-life VRSDP are indispensable for achieving low delivery cost and high quality service requirements from the enterprises, as well as the integration, rationalization, and standardization requirements from the administration.

The VRSDP is a complex combinatorial optimization problem (COP), in which computational cost increases exponentially with problem size, making it extremely difficult to find an optimal solution of a VRSDP in practical computational time. Simulated annealing (SA), genetic algorithms (GAs), tabu search (TS), and other methods have been proposed to find suboptimal solutions to VRSDPs [1, 2, 3, 4]. Since the 1990s, research on the VRSDP has expanded in Japan, with representative examples being [5] and [6, 7].

For the VRSDP, components of the evaluation function, i.e., evaluation criteria, e.g., running cost and loading ratio, etc. are mutually conflicting, in the sense that the running cost must be minimized and, at the same time, the loading ratio must be maximized. With such evaluation criteria, it is difficult for designers of real-world applications to set corresponding weights to reflect designers' intent.

In this invited talk, a concept of the vehicles dispatching problem for cooperative deliveries from multiple depots (VDP/CD/MD) is introduced. In order to solve the VDP/CD/MD, a practical calculation model of hierarchical multiplex structure is proposed. The proposed calculation model consists of 3 layers: atomic layer (system cost is controlled by a heuristic method), molecular layer (system state is adjusted by a heuristic method and an optimal calculation), and individual layer (a system plan is modified by fuzzy inference). The calculation model is implemented as a software component using object-oriented paradigm, and the corresponding optimization algorithm based on heuristic methods and fuzzy inference is also proposed. Experiments using the 3 days order data taken from an actual dispatching center in Tokyo area are done. A total of 27 tank lorries are available for daily cooperative deliveries from 3 depots to about 30-60 destinations. The transport area is in the Tokyo metropolitan area. Based on the experimental results, a detailed analysis is done from viewpoints of algorithm, system application, and practical implementation. The results and the evaluations by human experts confirm that the calculation model is a feasible, fast, efficient, and can be applied to the planning support system for the VDP/CD/MD in the real-world.

Since the calculation model and its algorithm take advantages of object-oriented modelling, heuristic method, and fuzzy inference, it can find a utility decision (vehicles plan) with intelligence and flexibility close to expert dispatcher. Because the vital input parameters are few and the computational engine is packaged into a software component, the calculation model is a convenient tool in system application for the VDP/CD/MD in the real-world. The proposed calculation model will be able to cover similar transportation problems in the real-world.

References

[1] I. Osman. *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, Annals of Operations Research, No. 41, pp. 421-451, 1993.

[2] F.Leclerc, J.Y.Potvin. *Genetic algorithm for vehicle dispatching* , Int. Trans. Opl.Res, Vol.4, No.5/6, pp. 391-400, 1997.

[3] G. Barbarosoglu, D. Ozgur. *A tabu search algorithm for the vehicle routing problem*, Computers & Operations Research, No. 26, pp. 255-270, 1999.

[4] C.D.Tarantilis, C.T.Kiranoudis. *A meta-heuristic algorithm for the efficient distribution of perishable food* , Journal of food Engineering, No. 50, pp. 1-9, 2001.

[5] H. Igarashi. *A vehicle schedule planning system using a simulated annealing method(in Japanese)* , Journal of the Transactions of the Institute of Electronics, Information and Communication Engineers, Japan, Vol. J78-D-II, No.5, pp. 819-826, 1995.

[6] K. Chen, Y. Takama, and K. Hirota. *A calculation model of hierarchical multiplex structure for vehicle routing & scheduling dispatching problem with single depot*, Japan Society for Fuzzy Theory and Systems, Vol.13, No.2, pp.187-198, 2001.

[7] F. Dong, K. Chen, E. M. Iyoda, H. Nobuhara, K. Hirota. *Solving Truck Delivery Problems Using Interated Evaluation Criteria Based on Neighborhood Degree and Evolutionary Algorithm* , J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.8, No.3, pp.336-345, 2004.

Kaoru Hirota, Fangyan Dong, Kewei Chen
Tokyo Institute of Technology
Department of Computational Intelligence & Systems Science
G3-49,4259Nagatsuta, Midori-ku, Yokohama 226-8502, Japan.
E-mail: hirota@hrt.dis.titech.ac.jp

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 55-56

Robust Statistical Translation Models: The Case for Word Alignment

Invited paper

Dan Tufiş

Abstract: The success of the statistical approaches in language processing has resurrected the interest in machine translation, which on its turn generated very useful results in many areas of language engineering. Within the architecture of a typical Statistical Machine Translation system, using a "noise-channel" paradigm, two basic information sources fundamentally influence the automatic translation accuracy: the translation model and the target language model.

The translation model encodes statistical information on how words or phrases from the source language are translated in the target language (including wording, local grouping of the translated words, part-of-speech mapping, etc). The references to target and source language texts are generic, identifying the direction of the processing in trying to reveal the equivalence relations over the two parts of a parallel text (bitext). The word alignment is an explicit representation of the pairs of words <wL1 wL2> (called translation equivalence pairs) co-occurring in corresponding parts of a bitext and representing mutual translations. The general word alignment problem includes the cases where words in one part of the bitext are not translated in the other part (these are called null alignments) and the cases where multiple words in one part of the bitext are translated as one or more words in the other part (these are called expression alignments).

The target language model encodes the distributional properties of the words in the output language (the grammatical ordering of the words depending on their part-of-speech (POS), idiosyncratic properties of the lexical items (such as case markers). The estimation of the accuracy of a language model is easily evaluated by a POS-tagging validation while for a translation model one could draw meaningful conclusions on the robustness of the model through a word-alignment exercise.

Neither POS-tagging nor word-alignment is an end in itself but necessary at one level or another to accomplish most natural language processing tasks. Because of this, is no surprise to see that the natural language research community invested and continue to invest a lot of energy in evaluating the progress in tagging and in word/phrase alignment. The technological competitions for integrated tasks, such as information retrieval, cross-language information retrieval, summarization, question-answering, machine translation, were systematically complemented by organized evaluations of the performances of the critical modules (tagging, chunking, parsing, name or time entity recognition, anaphora resolution, word aligning, word-sense disambiguation, etc).

The paper describes and evaluates our state-of-the-art word alignment system that combines two different word aligners, developed with independent motivations. The aligner combination achieves a significantly better result than each individual aligner does. We will report on the latest developments of our word-alignment system, winner of the second word-alignment competition organized in Ann Arbor, Michigan, USA, 2005. It implements a different approach from its predecessor, also a winner in the first word-alignment competition held in Edmonton, Canada, 2003.

References

[1] Brown, P. F., Della Pietra, S.A., Della Pietra, V. J., Mercer, R. L.1993. “The mathematics of statistical machine translation: Parameter estimation”. Computational Linguistics, 19(2) pp. 263-311.

[2] Dieterich, T., G. 1998. “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms”, Neural Computation, vol. 10, no. 7, pp. 1895-1923.

[3] Fan, R., Chen, P.H, Lin, C.J. 2005. “Working set selection using the second order information for training SVM”. (www.csie.ntu.edu.tw/ cjlin/papers/quadworkset.pdf). Technical Report, Dept. of Computer Science, National Taiwan University.

[4] Martin, J., Mihalcea, R., Pedersen, T. 2005. “Word Alignment for Languages with Scarce Resources”. In Proceeding of the ACL2005 Workshop on “Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond”. June, 2005, Ann Arbor, Michigan, June, Association for Computational Linguistics, 65-74.

[5] Melamed, D. 2001. Empirical Methods for Exploiting Parallel Texts. Cambridge, MA: MIT Press.

[6] Mihalcea R. Pedersen, T. 2003. “An Evaluation Exercise for Word Alignment”, in Proceedings of the HLT/-NAACL Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, Edmonton, Canada.

[7] Moore, R. 2002. “Fast and Accurate Sentence Alignment of Bilingual Corpora in Machine Translation: From Research to Real Users”. In Proceedings of the 5th Conference of the Association for Machine Translation in the Americas, Tiburon, California), Springer-Verlag, Heidelberg, Germany: 135-244.

[8] Och, F., J., Ney, H. 2000. “Improved Statistical Alignment Models”, Proceedings of ACL2000, Hong Kong, China, 440-447.

[9] Och, F.J., Ney, H. 2003. “A Systematic Comparison of Various Statistical Alignment Models”, Computational Linguistics, 29(1), pp. 19-51.

[10] Tufiş, D. Ion, R., Ceauşu, Al., Ştefănescu, D. 2006. “Improved Lexical Alignment by Combining Multiple Reified Alignments”. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL2006), Trento, Italy.

[11] Tufiş, D., Ion, R. Ceauşu, Al., Stefănescu, D. 2005. “Combined Aligners”. In Proceeding of the ACL2005 Workshop on “Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond”, Ann Arbor, Michigan, June, Association for Computational Linguistics, pp. 107-110.

[12] Tufiş, D., Ion, R., Ide, N. 2004. “Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets”. In Proceedings of the 20th International Conference on Computational Linguistics, COLING2004, Geneva, pp. 1312-1318

[13] Tufiş, D., Barbu, A., M., Ion, R. 2003. “A word-alignment system with limited language resources”. In Proceedings of the NAACL 2003 Workshop on Building and Using Parallel Texts; Romanian-English Shared Task, Edmonton, 36-39.

[14] Tufiş, D. 2002. “A cheap and fast way to build useful translation lexicons”. In Proceedings of the 19th International Conference on Computational Linguistics, COLING2002, Taipei, pp. 1030-1036.

Dan Tufiş
Research Institute for Artificial Intelligence
of the Romanian Academy
E-mail: tufis@racai.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 57-61

Control Systems Modelling and Design for Processes Synchronization

Victor Ababii, Viorica Sudacevschi, Emilian Guţuleac

Abstract: Design of real-time control systems requires new methodologies in their modelling, verification and implementation. In the paper a complex integrated design system is presented. The design process starts with analysing of the Petri net model of the control system in a special software environment that performs modelling, verification, validation and performance evaluation of the model, its conversion into AHDL code (Hard Petri net), simulation of the obtained code in MAX+ Plus II environment and FPGA or CPLD configuration of the control system.

Keywords: HDL Design, Petri Net Models, Hardware Implementation, Performance evolution, System Control.

1 Introduction

The increasing complexity of real-time control systems requires new approaches for their modelling, synthesis and verification. A lot of scientific research studies propose different methods for design stages integration into an automatic flow with minimal human participation [1, 2, 3, 4]. These methods are based on Petri net type models as the first step in control system design and their conversion into a program code that is executed on PLC systems. Other research direction is the Petri net model implementation in basic logic elements that can be used in control systems or in modelling systems [5, 6, 7]. In this paper is presented an integrated design environment that support synthesis, modelling and validation of a control system with concurrent data processing based on Petri net model, performance analysis and translation of this model into AHDL code that allows control system configuration into FPGA or CPLD circuits.

2 Diagram of synthesis flow

Control system synthesis is executed according to the diagram that is presented in Figure 1. Synthesis stages description:

PN Models Source - Petri net model that is proposed for analysing is introduced in graphical form;

VPNP Tool - software tool that allows inserting and modifying in an interactive mode the Petri net model;

Analysis (**Reachability graph and Structural analysis**) - The proposed Petri net model is analysed in order to determine the set of reachable states and to form the reachability graph. The structural analysis determines the main properties of the model such as its safety and viability;

MI and MO generation - incidence matrix and initial marking generation and their storage in corresponding files;

HDL Compiler - AHDL code compilation based on matrixes $*.imf$ and $*.rag$;

HDL Objects Library - the library with standard AHDL objects that are used to form AHDL code of a Petri net;

HDL code - the obtained after compilation AHDL code;

Max Plus + II Design Tool - MAX+PLUS II software is a fully integrated, architecture-independent package for designing logic with ALTERA programmable logic devices;

FPGA or CPLD Device - FPGA or CPLD configuration of the Petri net model.

3 The VPNP Tools

The interactive environment VPNP represents a software tool with a graphical interface, designed for Petri net models analysing. It allows to draw a graphical Petri net model, to store into a file and to read from a file these models and to perform the structural analysis of the models with visualization of the results [8]. After analysing of the Petri net model the incidence matrix ($*.imf$) and initial state ($*.rag$) files are obtained.

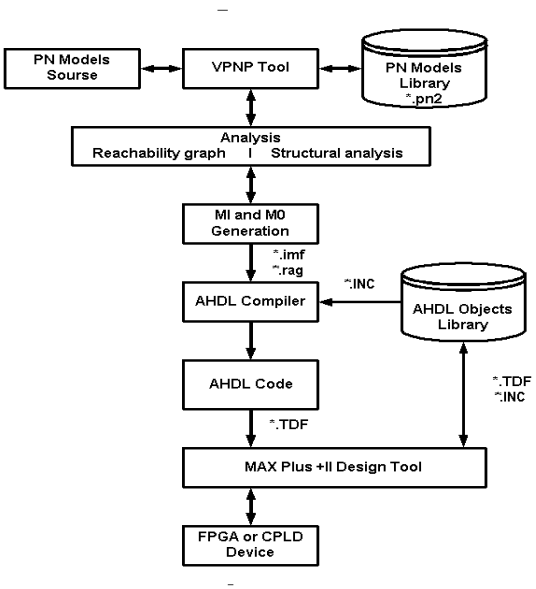


Figure 1: Diagram of synthesis flow

4 Petri net model for hardware implementation

A Petri net(PN) is a 5-tuple:$PN =< P,T,W^+,W^-,M_0 >$,where:

$P = \{p_i, i = \overline{1,I}\}$-is a finite and non-empty set of places; $T = \{t_j, j = \overline{1,J}\}$- is a finite and non empty set of transitions; $W^+ = \{(p_i,t_j)\}$- is a set of arcs from place p_i to transition t_j; $W^- = \{(t_j,p_i)\}$- is a set of arcs from transition t_j to place p_i ; $M_0 = \{m_1,m_2,...,m_I\}$- is the initial marking.

The Petri net changes its states according to functional rules that are defined for each class of Petri nets [13, 14, 3]. The architecture of the system with concurrent data processing represents a set of processor elements with data flow interconnections [12]. For a Petri net model data flow will depend on the internal structure of the net model. Taking this into consideration, we can define a Hardware Petri Net ( ) as a set of processor elements (transitions and places) and data flows (arc connections):

$HPN = \{T \cup P \cup A^+ \cup A^-\}$, where:

$T = \{t_1,...,t_J\}$- is set of transition type processor elements;

$P = \{p_1,...,p_J\}$- is a set of place type processor elements;

$A^+ = \{A_1^+,A_2^+,...,A_J^+\}$- is a set of increment connections to each place, where:

$$A_j^+ = \begin{cases} a_{j,i}^+ = 1 & \text{if exists a connection between } t_j \text{ and } p_i, \\ a_{j,i}^+ = 0 & \text{if do not exist a connection between } p_i \text{ and } t_j. \end{cases}$$

$A^- = \{A_1^-,A_2^-,...,A_J^-\}$- is a set of decrement connections from each place, where:

$$A_i^- = \begin{cases} a_{i,j}^- = 1 & \text{if exists a connection between } p_i \text{ and } t_j, \\ a_{i,j}^- = 0 & \text{if do not exist a connection between } t_j \text{ and } p_i \ . \end{cases}$$

Incidence matrix is obtained as: $IM = A^+ \cup A^-$. The pair (m_i,P_i) determines the state of the processor element P_i. The set of all states $S^k = \{(t_j,P_i), \forall i = \overline{1,I}\}$ for places determines the global state of the system at k iteration, where, $k \in K$. The state $S = \cup_{k=1}^{K} S^k$ determines the set of allowed states for the system and the reachability graph for the Petri net model.

5 Processor elements specification

The hardware implementation of Petri Net contains two main parts: the processor element transition (T) and the processor element place (P).

The processor element Transition (T) prepares the data processing operation. After global state $S^k = \{(t_j, P_i), \forall i = \overline{1,I}\}$ analysing at the step of data processing, the condition for step $k+1$ of data processing operation is formed.

The logic symbol of a functional element transition is presented in Figure 2a, where: $CLC(C)$- clock signal; Inc_{ij} - increment outputs connected to all output places to this transition; Dec_{ij} - decrement outputs connected to all input places to this transition, S_j^k - Petri net model state signal inputs for transition T_j. For all transitions the logic function $Inc_{ij} = Dec_{ij} = \prod_{m=1}^{M}(s_{jm}^k)$ is formed that allows the transition to fire only if all inputs $m = 1,...,M$ will have the logic value "1".

The processor element Place P stores the state value and performs the increment and decrement operation of the number of tokens. The logic symbol for processor element Place P is shown in Figure 2b, where: $CLC(C)$- clock signal; Inc_{ij} - enable inputs for increment operation of the number of markers in place; Dec_{ij}- enable inputs for decrement operation of the number of markers in place; S_j^k- place state at the k iteration step that determines the marking presence in place. The number of tokens in place $m_i^{k+1}, i = \overline{1,I}$ is changed according to the following formula:

$$m_i^{k+1} = \begin{cases} 1 & \text{if} \quad \sum_j^J (Inc_{ij}) = 1 \wedge (m_i^k) = 1 \vee (m_i^k) = 1 \\ 0 & \text{if} \quad \sum_j^J (Dec_{ij}) = 1 \wedge (m_i^k) = 1 \\ m_i^k & \text{if} \quad \sum_j^J (Inc_{ij}) = 0 \wedge (m_i^k) = 0 \\ m_i^k & \text{if} \quad \sum_j^J (Inc_{ij}) = 1 \wedge (m_i^k) = 1 \end{cases}$$

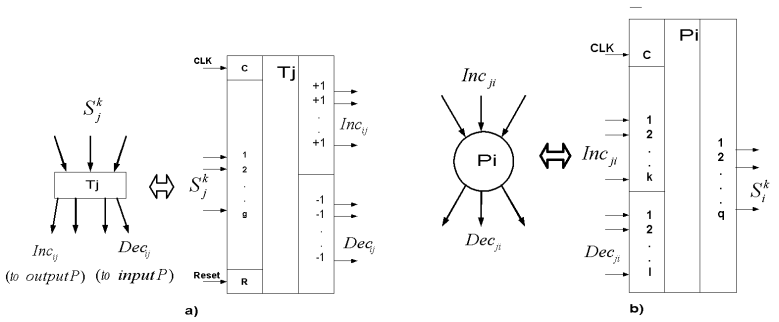


Figure 2: Functional element (a) Transition and (b) Place

6 HDL Compiler

A HDL compiler performs conversion of the Petri net model that is defined by the incidence matrix IM (file $*.imf$) and initial state matrix $M0$ (file $*.rag$), obtained in VPNP software tool, to AHDL code. The dialog window of the software tool HDLCS allows to insert the incidence matrix IM (command OPEN), the initial marking M0 (command LOAD M0) and to save the AHDL code of the Petri net model. Command PROCESS starts the compilation. The AHDL code is obtained after processor elements selection from HDL Library according to their characteristics, their interconnections according to the incidence matrix IM and generation of the file that contains the source AHDL code with TDF extension. At the first step of AHDL code generation in the file are included processor elements Place and Transition, the global synchronization clock is defined, and interconnections between processor elements are formed.

7 An Example of Control System for Synchronisation of Data Communication

The proposed method was used to design a control unit for data transfer in a computer system. The structure of the control unit is presented in figure 3a. Where the source and destination block communicate using following signals: STB - Strobe when sending byte by Data Bus and ACK signal to confirm the data reception.The Petri net model for control system modelling and control unit implementation is shown in figure 3b and figure 3c,

respectively. After VPNP structural analysis of Petri net model the incidence matrix and initial marking (M0 = [0,0,0,0,1]) where obtained.

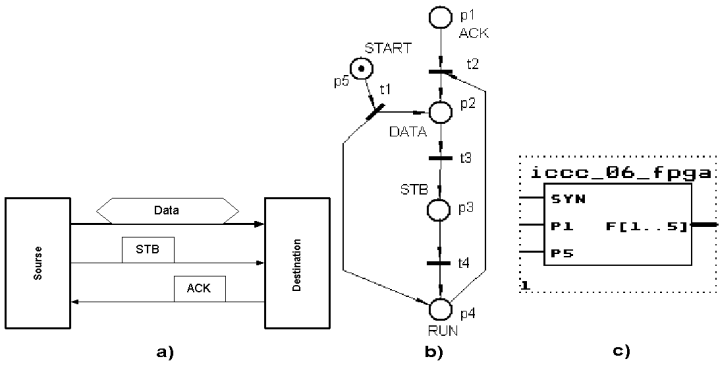


Figure 3: The structure of Control Unit (a) Petri net model for implementation (b) and the Control Unit

Table 1. The AHDL code

```
INCLUDE "place2x1.inc";
INCLUDE "transition1.inc";
INCLUDE "place1x1.inc";
INCLUDE "transition2.inc";
SUBDESIGN icccc_06_fpga
(
        SYN: input;
        p1 : input; %ACK%
        p5 : input; %START%
        f[1..5] : output; %f[3]  -STB%
)
VARIABLE
        p2: place2x1;
        p3: place1x1;
        p4: place2x1;
        t2: transition2;
        t3: transition1;
        t1: transition1;
        t4: transition1;
BEGIN
        t4.Min0=p3.Mout;
        t4.SYN=!SYN;
         t2.Min0=p1; %.Mout; %
        t2.Min1=p4.Mout;
        t2.SYN=!SYN;
         t3.Min0=p2.Mout;
        t3.SYN=!SYN;
        t1.Min0=p5; %.Mout; %
        t1.SYN=!SYN;
        p2.Inc0=t1.Dec_Inc;
        p2.Inc1=t2.Dec_Inc;
        p2.Dec0=t3.Dec_Inc;
        p2.SYN=SYN;
        p2.nCLR=VCC;
        p2.nPR=!(!p5 & GND & !SYN);
                                %Set M0%
        f[2]=p2.Mout;
        p3.Inc0=t3.Dec_Inc;
        p3.Dec0=t4.Dec_Inc;
        p3.SYN=SYN;
        p3.nCLR=VCC;
        p3.nPR=!(!p5 & GND & !SYN);
                                %Set M0%
        f[3]=p3.Mout;
        p4.Inc0=t1.Dec_Inc;
        p4.Inc1=t4.Dec_Inc;
        p4.Dec0=t2.Dec_Inc;
        p4.SYN=SYN;
        p4.nCLR=VCC;
        p4.nPR=!(!p5 & GND & !SYN);
                                %Set M0%
        f[4]=p4.Mout;
END;
```

The AHDL code is presented in table 1. The code is processed in MAX + Plus II tool. The obtained control unit for data transfer is presented in figure 3c, where: SYN - synchronization signal; P1 - ACK input generated by the destination object; P5 -input for data transfer initialisation; F[2] -transfer to data bus; F[3] - generates STB signal for data transfer; F[4] - the system is ready for the next operation. The statistic report obtained after FPGA compilation shows that it was used 12 LCs. The timing diagrams obtained after simulation confirm the correctness of control system functionality.

8 Conclusions

We have described the design of a control system using Petri nets. The proposed integrated system uses Petri net model for modelling and verification of control system functionality, conversion of the model into HDL code and its implementation into FPGA or CPLD circuits. The proposed method allows a high flexibility in quick reconfiguration of control algorithms. The obtained results prove the reliability of the integrated system.

We plan to continue investigation of this method. One of the most important research directions is the concurrent data processing analysis, new synchronization methods for data processing operations, functional extension of the processor elements for Timed Petri net implementation.

References

[1] J. M. Fernandes, A. J. Proeneca,M. A. Adamski, “VHDL Generation from Petri Net Parallel Controller Specification,” *ser. In Proceeding of EUROVHDL’ 95*, Brighton, GB, 18.-22.09, 1995.

[2] W. Fengler, A.Wendt, M.A. Adamski, J. L. Monteiro, “Net Based Program Design for Controller Systems,” *In Proceeding of 13th IFAC World Congress*, San Francisco, June 30 - July 5, 1996.

[3] A.H. Jones, M. Uzam, A.H. Khan, D. Karimzadgan, S.B.Kenway, “A General Methodology for Converting Petri Nets Into Ladder Logic: The TPLL Methodology,” *In Proceedings of the 5th International Conference on Computer Integrated Manufacturing and Automation Technology - CIMAT’96*, May, France, pp. 357-362, 1996.

[4] A.H. Jones, M. Uzam, N. Ajlouni, “Design of Discrete Event Control Systems for Programmable Logic Controllers Using T-timed Petri Nets,” *In Proceedings Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design - CACSD’96*, Dearborn, MI, USA, September 15 - 18, pp. 212 - 217, 1996 .

[5] Murat Uzam, Mutlu Avci and M. Kursat Yalcin, “Digital Hardware Implementation of Petri Net Based Specifications: Direct Translation from Safe Automation Petri Nets to Circuit Elements,” *In Proceedings of the International Workshop on Discrete-Event System Design, DESDes’01*, June 27-29, Przytok near Zielona Gora, Poland, 2001.

[6] G. A. Bundell, “An FPGA implementation of the Petri Net firing algorithm,” *In Proceedings of the 4th Australasian Conf. on Parallel and Real-Time Systems*, pp. 434-445, 1997.

[7] John Morris et al., “A Re-configurable Processor for Petri Net Simulation, ” *In Proceedings of the 33rd Hawaii International Conference on System Sciences*,2000.

[8] E.Gutuleac, A. Reilean, C. Bosneaga, “Visual Petri Net plus -integrate Program Package for modeling using Stochastic Petri Nets,” *Proceedings of the 3-rd International Conference on Information Technologies-2001, BIT+2001*, 11-13 April, Vol.1, Chisinau, Moldova, Page 46, 2001.

[9] J. Peterson, “Petri Net theory and the modeling of systems,” New-York, 1984.

[10] T. Murata, “Petri Nets: Properties, Analysis and Applications,” *In Proceeding of the IEEE, vol. 77, no. 4*, pp. 541-580, 1989.

[11] E. Gutuleac, “Modelarea si evaluarea performantelor sistemelor de calcul prin retele Petri,” Partea I, DEP UTM, Chisinau, 1998.

[12] D. Culler, J. P. Singh, “Parallel Computer Architecture, Morgan Kaufmann, ISBN 0-678-954-341-2, 1999.

Victor Ababii,
Viorica Sudacevschi,
Emilian Guţuleac
Technical University of Moldova
Computer Science Department
Address: 168, Bd. Stefan cel Mare, MD-2004
Chişinău, Republic of Moldova
E-mail: {avv, svm, egutuleac}@mail.utm.md

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 62-67

On Some Methods for Non-Stationary Time Series Analysis: a Java-based software

Grigore Albeanu

Abstract: The modelling and analysis of time series are important actions for a large spectrum of applications. This paper considers stationary and non-stationary time series from computational point of view. Some existing techniques are reviewed for probabilistic, wavelet-based, soft computing and hybrid approaches. The functionalities of the *TJmes* - a Java-based software tool for time series modelling and analysis - are described both for computing and E-learning in the time series field.

1 Introduction

A time series is a sequence of observations, ordered in time or space, which provides useful information about the evolution of some physical, biological, social or economic systems. To illustrate such an importance let us mention some usage examples: econometry [7] (stock exchange indices, profits, imports, exports etc.), sociology (crime rate, unemployment, different enrollments etc), meteorology [4] (temperature, wind speed, rainfall etc), hydrology, environment pollution, epidemiology, sciences [14, 13, 26, 27] etc. From theoretical point of view a time series is a random process [26]. A standard graphical representation for time series is shown in figure 1.

When observations (denoted by x(t)) are made at every moment of time then a *continuous* time series is obtained. In the case of discrete moments (usually equi-spaced, and denoted by x[t]: t = 0, 1, ..., N-1) the time series is called to be *discrete*. When only one variable is measured over time, we deal with *univariate* time series; otherwise, the time series is *multivariate*.

Four *characteristics* can give a complete understanding of the time series: the trend, the seasonal, the cyclic and the irregular component. Some applications are interested in trend forecasting, but others need information about the periodicity of the analysed process.

There are two main approaches to time series analysis. The first one (called the *time domain* approach) represents time series as a function of time and is useful to obtain the trend component and, then, to propose a forecasting model. The second approach deals with the *frequency domain*, in order to determine the periodic components of the time series.

Stationary and *nonstationary* time series are used to model real phenomena. Stationarity of time series implies the homogenity of the series, that means the series behaves in a similar way regardless the time sampling. Mathematically, the stationarity implies the invariance of the joint probability distribution of the process under observation. For some practical situations the following requirements are needed: the mean and variance are constant over the time, but the autocovariance of every two elements depends only on their temporal separation.

However, "many phenomena, both natural and human influenced, give rise to signals whose statistical properties change under time translation" as mentioned in [11]. Such phenomena, called *nonstationary*, are difficult to investigate and numerous methods have been proposed, including wavelets [3, 8, 13, 18], soft computing techniques [1, 4, 12, 16, 20] and hibrid approaches [16, 22, 24], to mention only some references.

In this paper we investigate different approaches and propose a distributed architecture for computer aided time series modelling and analysis. Also, the software tool integrates an E-learning module to be used for data mining lessons. In the second section we cover some probabilistic approaches. Wavelet-based modelling and analysis are considered in the third section. Soft computing methods, and hybrid methods are covered in the fourth, respectively the fifth section. The final section discuses the software requirements of a Java-based software tool both for time series modelling-analysis-forecasting and e-learning.

2 Probabilistic time series modelling and analysis

An interesting feature of discrete univariate time series can be described using autocovariance and autocorrelation functions. The main difference between deterministic data and random time series shows the persistence of the autocorrelation functions along the time displacements in the deterministic case. Let $k \geq 1$, and r_k be the k-th order correlation coefficient between observations separated by k time units. The array of autocorrelation coefficients r_k,

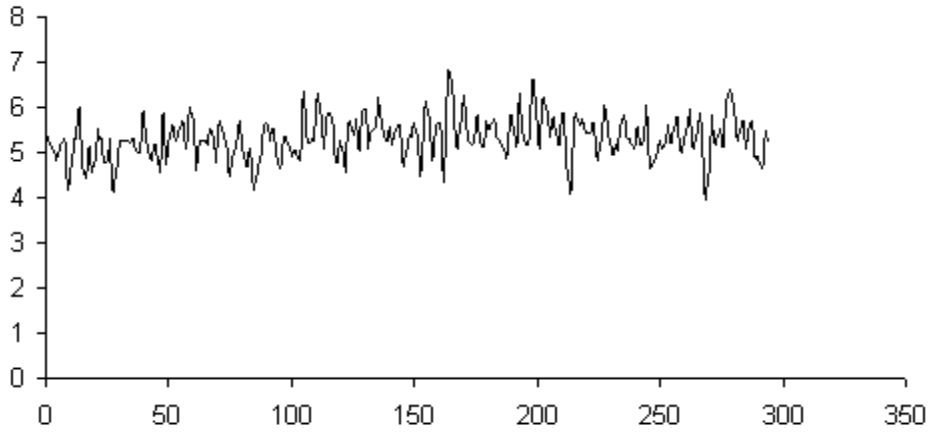


Figure 1: A standard graphical representation

plotted with k as abscissa and r_k as ordinate formed the so-called a *correlogram* that provides useful information to identify the type of time series according to the following rules:

- If a time series is completely random, then for large N, $r_k = 0$ for all k.
- The stationary time series have short term correlations.
- If succesive values of a time series tend to alternate, the correlogram would also tend to alternate.
- For stationary time series, when the time series has a trend, the values of r_k would not decrease to zero, except for very large values of k. This is not the case for nonstationary time series.
- If a time series is characterized by seasonal fluctuations, then the correlogram would also shows oscillations along the same period.

To measure the similarity between two time series the cross-corelation (i.e. the linear correlation coefficient between two time series) can be used, also based on the size of time lag. Other analysis can be realized using the initial time series [1, 9, 23, 26], transformed series [2, 3, 5, 10, 14], or based on some features (patterns) already extracted [11, 17, 19, 24].

Any computer aided tool for time series modelling and analysis should consider a module for pre-processing, that means a component to transform the data to be suited for some algorithm. As an example, sometimes is necessary to remove the trend (by subtracting the fitted time series model from the original time series) and smoothing the time series in order to stabilize the variance. For trend removing, also, can be used first-differencing between consecutive values, filtering and trend identification. When stationary time series, for trend studying we can use regression models, moving average models, autoregressive models (AR), autoregressive moving average models (ARMA) and state-space models [23, 26]. In general, is necessary to select an appropriate model based on some criteria ([23], pag. 178). To study adaptive AR and ARMA models the following algorithms can be used: the Recursive Least-Squares algorithm (RLS) and the square-root RLS, the Extended Least-Square algorithm (ELS), the Recursive Maximum Likelihood (RML) algorithm, stochastic gradient algorithms based on Gauss-Newton method, Lattice type algorithms based on projection operators etc.

When studying time series in the frequency domain, spectral analysis is necessary to decompose a stationary time series into a sum of components from an adequate function space. Let us mention the very used Fourier transform (including the Fast Fourier Transform) and the spectrum estimation by parametric (Yule-Walker autoregressive method , Burg method), nonparametric (Periodogram and the Welch's method), and subspace methods (based on eigen-processing of the correlation matrix). Filtering is based on low, high or band pass filters. The extension of the cross-corelation method to the frequency domain is called *cross-spectrum* and it is useful for study the relationship between any two time series, and estimate the coherency between the series.

3 Wavelet-based time series modelling and analysis

According to [13], "wavelets are mathematical functions that cut up data into different frequency components and then study each component with a resolution matched to its scale." The extension from Fourier analysis to wavelet-base modelling and analysis is obtained by starting with the Harr function ψ, given by

$$\psi(x) = \begin{cases} 1, & 0 \le x < \frac{1}{2} \\ -1, & \frac{1}{2} \le x < 1 \\ 0, & \text{otherwise} \end{cases}$$

and called a *mother wavelet* because it is used to generate a large family of wavelets by means of dyadic dilations and integer translations. If j is the dilation index and k is the translation index, then a complete orthonormal system for $L^2(R)$ denoted by $\{\psi_{j,k}, j,k \in Z\}$ can be obtained if

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k).$$

However, a wavelet-based analysis is done using a scaling function $\phi(t)$ that satisfies an equation such as

$$\phi(t) = \sum_{k \in Z} c_k \phi(2t - k),$$

where the family $\phi(t-k)_{k \in Z}$ forms an orthogonal set of functions.

A discrete method of finding a basis is needed because most data sets are values corresponding to a finite number of discrete time points. Let us denote by N the number of observations and by J the integer number such as $N = 2^J$. If the observed values are denoted by $X = (x(t_0), x(t_1), \ldots, x(t_{N-1}))^T$, then the time series model can be described as:

$$X(t_i) = \sum_{j=1}^{J-1} \sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t_i) + c_{0,0}\phi(t_i),$$

using the so-called *discret wavelet transform* (DWT), where ϕ is the scaling function (father wavelet) associated with ψ. The coefficients $d_{j,k}$ are given by $d_{j,k} = \sum_i x(t_i)\psi_{j,k}(t_i)$.

The above ideas can be used for the detection and estimation of the trend component of the time series. Other wavelet tools (non-decimated wavelet transform, wavelet packets) are also very useful for time series modelling and analysis as shown in [13, 18]. Following [14], let us mention the following advantages obtained by using wavlet packets:

- Wavelet packets belong to well organized collections, and every collection is an orthogonal basis for $L^2(R)$.
- The best model can be selected by comparisons between at least two wavelet packet representations.
- A simple algorithm is available for supporting wavelet packets.
- Some classical wavelets are particular cases of wavelet packets (for instance the Daubechies's orthogonal wavelets).

>From these points of view, we ask for a wavelet-base modelling and analysis module to be considered for design and implementation into the TJmes software. The *piramidal* algorithm will be implemented for obtaining DWT.

4 Soft computing techniques for time series modelling and analysis

Soft computing is tolerant of imprecision, uncertainty, partial truth, and approximation. Fuzzy Logic and Probabilistic Reasoning based on knowledge-driven reasoning, also called *approximate reasoning*, Neural Computing and Evolutionary Computation as *data-driven search* and *optimization approaches* - are the principal constituents of the soft computing field. These constituents, which are complementary rather than competitive, can be used in time series modelling and analysis.

One approach in fuzzy time series prediction uses all available input-output data pairs in order to build a rule base [1, 27]. Let us consider a fuzzy system with inputs $(x_0, x_1, ..., x_{M-1})$, an output y, and N data points in the training set. For systems having more inputs, the procedure will work in a similar manner. In order to create fuzzy rules with fixed membership functions, five steps are necessary to be followed [27]:

- Define the fuzzy partitions for the input and output variables.
- Generate one fuzzy rule for each input-output pair, and obtain an initial fuzzy rule-base.
- Calculate the membership degree D for each fuzzy rule belonging to the fuzzy rule-base created in the previous step.
- Removing inconsistent and redundant rules [27] and, create the final fuzzy rule-base. A reliability factor (RF) [15] can be computed for every set of K rules with the same antecedent part by the ratio $K1/K$, where $K1$ is the number of redundant rules. In this manner, for each rule, an effective degree ED can be obtained as in formula: $ED = D * RF$. Finally, the rule-base will contain fuzzy rules with the largest effective degrees.
- Select the inference scheme and perform the fuzzy inference procedure [27]. Select a defuzzification scheme to provide a crisp value as output. A variety of defuzzification schemes can be found in [27].

The simplest fuzzy rule-base has rules of the form **IF** X(t) is A **THEN** X(t+1) is B, where A and B are fuzzy sets. However, the above approach can include more information in the antecedent part and complex rules can be created. Another approach consists of exploring a nearest neighbour pattern for time-series forecasting in a fuzzy manner, called the fuzzy nearest neighbour method [25].

Sometimes it is necessary to extract information from time series and use it in different statistical-based data analysis. Let us mention the approaches given in references [4, 12, 20].

Let us mention also the usage of neural networks, genetic algorithms and hybrid soft computing approaches for data discovery from time series. Neural networks [6], and combined neuro-fuzzy networks and genetic algorithms [28, 29] are used for both univariate and multivariate time series modelling and analysis. A fuzzy-wavelet prediction method is described in [22]. The following modules will be built-in TJmes software: time series data base management, rule-base management and soft computing data mining.

5 A Java-based software tool for time series modelling and analysis

The computer aided time series modelling and analysis, called TJMES (**T**i**ME** serie**S** by **J**ava), will implement methods for processing large databases containing information that can be interpreted as time series. Computational statistical methods, wavelet-based algorithms and soft computing techniques will be availlable to the data analyst in a user-friendly environment implemented completely in Java.

For non-stationarity testing, both formal methods (computer intensive) and Partial Autocorrelation Function Charts (graphical statistics) will be implemented. A regression method can be used for processing the transformed time series after the non-stationary elimination. A module for graphical representations will be designed and integrated in TJmes

A time series file format will be designed based on XML specification in order to obtain full software interoperability. Also, the software tool will integrate an E-learning module to be used for on-line data mining lessons.

6 Summary and Conclusions

Time series are used for modelling and analysis of many real life phenomena. A platform independent software tool for time series modelling and analysis is a requirement. In this project, we investigate the main approaches on time series data mining and forecasting, and software requirements for a software tool useful for modelling, analysis, and computer aided learning are established. The TJmes software is under development and preliminary results will be available in short time.

Acknowledgements. These investigations were supported by UNESCO IT Chair at University of Oradea in the framework of the "Advanced ITC Methodologies" research program.

References

[1] G. Albeanu & Fl. Popentiu-Vladicescu, "*On using the fuzzy nearest neighbour method for time series forecasting in software reliability*,", Proc. of SIG'2005 Conference, Sinaia, Romania,

[2] A. Antoniadis, G. Gregoire & I. McKeague, "*Wavelet methods for curve estimation*, " JASA, Vol. 89, pp. 1340-1353, 1994.

[3] S. Van Bellegem, P. Fryźlewicz & R. von Sachs, "*A wavelet-based model for forecasting non-stationary processes*," in J-P. Gazeau, R. Kerner, J-P. Antoine, S. Metens, J-Y. Thibon (Eds.), *GROUP 24: Physical and Mathematical Aspects of Symmetries*, IOP Publishing, Bristol, http://www.maths.bris.ac.uk/˜mapzf/g24/g24.pdf, 2003.

[4] M. R. Berthold, M. Ortolani, D. Patterson, F. Höppner, O. Callan & H. Hofer, "*Fuzzy information granules in time series data*," International Journal of Intelligent Systems, Vol. 19, No. 7 , pp. 607 - 618, 2004.

[5] B. Burtschy, G. Albeanu, D. N. Boros, Fl. Popentiu & V. Nicola, "*Improving Software Reliability Forecasting*," Microelectronics and Reliability, Vol. 37, No. 6, pp. 901-907, 1997.

[6] S. Canu, X. Ding & Y. Grandvalet, "*Une application des réseaux de neurones pour la prévision à un pas de temps*," in Thiria S., Lechevallier Y., Gascuel O. & Canu S. (Eds), *Statistique et méthodes neuronales*, chapitre 7, pp. 120-131, Dunod, 1997.

[7] C. Wai Cheong, W. Wei Lee & N. A. Yahaya, "*Wavelet-based temporal cluster analysis on stock time series*," ICOQSIA 2005, 6-8 December, Penang, Malaysia, 2005.

[8] P. F. Craigmile & D. B. Percival, "*Wavelet-Based Trend Detection and Estimation*," in A. H. El-Shaarawi & W. W. Piegorsch, Chichester (Eds.) *Encyclopedia of Environmetrics* (Volum 4), John Wiley & Sons, England, pp. 2334-2338, http://faculty.washington.edu˜dbp/PDFFILES/trendencyclo.pdf, 2002.

[9] R. Dahlhaus, "*Fitting Time Series Models to Nonstationary Process*," Annals of Statistics, Vol. 25, pp. 1-37, 1997.

[10] P. Fryzlewicz, S. Van Bellegem, & R. von Sachs, "*Forecasting non-stationary time series by wavelet process modelling*," Annals of the Institute of Statistical Mathematics, Vol. 55, pp. 737-764, http://www.maths.bris.ac.uk˜mapzf/flsw/pred_lsw.pdf, 2003.

[11] K. Fukuda, H. Eugene Stanley & Luís A. Nunes Amaral, "*Heuristic segmentation of a nonstationary time series*," Phys. Rev. E 69, 021108 (12 pages), http://amaral.chem-eng.northwestern.edu/Publications/Papers/Fukuda-2004-Phys.Rev.E-69-021108.pdf, 2004.

[12] S. Giove, "*Fuzzy logic and clustering methods for time series analysis*," ESIT'99, Workshop on Finance, Trade and Services, Chania, http://www.erudit.de/erudit/events/esit99/12600_p.pdf (seven pages), 1999.

[13] D. A. Goodwin, S. Barber, & R. G. Aykroyd, "*Wavelet packet modelling of seismic data*," in R.G. Aykroyd, S. Barber, & K.V. Mardia (Eds.), *Bioinformatics, Images, and Wavelets*, pp. 126-129, http://www.maths.leeds.ac.uk/Statistics/workshop/lasr2004/Proceedings/goodwin.pdf, Department of Statistics, University of Leeds, 2004.

[14] S. Jaffard, Y. Meyer & R. D. Ryan, "*Wavelets. Tools for Science & Technology*," SIAM, Philadelphia, 2001.

[15] X. Liu, B. K. Kwan & S. Y. Foo, "*Time Series Prediction Based on Fuzzy Principles*," Preprint, Department of Electrical & Computer Engineering, Florida State University, 2003.

[16] H. Madsen, P. Thyregod, B. Burtschy, G. Albeanu & F. Popentiu, "*On Using Soft Computing Techniques in Software Reliability Engineering*," International Journal of Reliability, Quality, and Safety Engineering, Vol. 13, No. 1, pp. 1-12, 2006.

[17] J. P. Morrill, "*Distributed Recognition of Patterns in Time Series Data*," Communications of the ACM, Vol. 41, No. 5, pp. 45-51, 1998.

[18] G. P. Nason, T. Sapatinas, & A. Sawczenko, "*Wavelet packet modelling of nonstationary wind energy time series*," Technical Report, Department of Mathematics, University of Bristol, Bristol, http://www.maths.bris.ac.uk/˜magpn/Research/unpubpapers/wpm.pdf, 1997 (Revised July, 1999.)

[19] R. Todd Ogden, "*Essential Wavelets for Statistical Applications and Data Analysis*," Birkäuser, Boston, 1997.

[20] M. Ortolani, H. Hofer, D. Patterson, R. Höppner & M. R. Berthold, "*Fuzzy information granules in time series data*," Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Vol. 1, pp. 695-699, 2002.

[21] Fl. Popentiu-Vladicescu, B. Burtschy & G. Albeanu, "*Time series methods for modelling software quality*," in E. Zio, M. Demichela, N. Piccinini, *Towards a safer world, Proceedings of the European Conference On Safety and Reliability*, ESREL 2001, Politechnico Di Torino, Italy, Vol. 1. pp. 9-15, 2001.

[22] A. Popoola, S. Ahmad & K. Ahmad, "*A Fuzzy-Wavelet Method for Analyzing Non-Stationary Time Series*, " in *Proc. of The 5th Int. Conf. on Recent Advances in Soft Computing* (December 16-18, 2004, Nottingham, UK), http://www.computing.surrey.ac.uk/grid/fingrid/papers_files/Reports/12.pdf, 2004.

[23] B. Porat, "*Digital Processing of Random Signals. Theory & Methods*," Prentice-Hall, New Jersey, 1994.

[24] R. J. Povinelli & Xin Feng, "*Data Mining of Multiple Nonstationary Time Series*," Artificial Neural Networks in Engineering, Proceedings, pp. 511-516, http://povinelli.eece.mu.edu/publications/papers/annie1999b.pdf, 1999.

[25] S. Singh, "*Fuzzy Nearest Neighbour Method for Time-Series Forecasting*," Proc. 6th European Congress on Intelligent Techniques and Soft Computing, Vol. 3, pp. 1901-1905, 1998.

[26] M. Tertisco, P. Stoica & Th. Popescu, "*Modelling and forecasting of time series*," Romanian Academy Press (in romanian), Bucharest, 1985.

[27] I. Văduva & G. Albeanu, "*Introduction to fuzzy modelling*," Bucharest University Press (in romanian), 2003.

[28] J. Valdés & G. Mateescu, "*Time series model mining with similarity-based neuro-fuzzy networks and genetic algorithms: a parallel implementation*," Proceedings of the RSCTC'02, Special Session on Distributed and Collaborative Data Mining, Pensylvania, National Research Councel of Canada (44933), 2002.

[29] J. Valdés & G. Mateescu, "*Multivariate Time Series Model Discovery with Similarity-Based Neuro-Fuzzy Networks and Genetic Algorithms*," Proceedings of the IEEE, INNS, IJCNN 2003 International Joint Conference on Neural Networks, IEEE Catalog Number: 03CH37464C, 2003.

Grigore Albeanu
University of Oradea
UNESCO Chair in Information Technologies
Address: 1, University Street, 410087, Oradea, Romania
E-mail: galbeanu@netscape.net

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 68-73

From Algorithms to (Sub-)Symbolic Inferences in Multi-Agent Systems

Boldur E. Bărbat, Sorin C. Negulescu

Abstract: Extending metaphorically the Moisilean idea of "nuanced-reasoning logic" and adapting it to the e-world age of Information Technology (IT), the paper aims at showing that new logics, already useful in modern software engineering, become necessary mainly for Multi-Agent Systems (MAS), despite obvious adversities. The first sections are typical for a position paper, defending such logics from an *anthropocentric perspective*. Through this sieve, Section 4 outlines the features asked for by the paradigm of *computing as intelligent interaction*, based on *"nuances of nuanced-reasoning"*, that should be reflected by agent logics. To keep the approach credible, Section 5 illustrates how *quantifiable synergy* can be reached - even in advanced challenging domains, such as *stigmergic coordination* - by injecting symbolic reasoning in systems based on sub-symbolic "emergent synthesis". Since for *future work* too the preferred logics are doxastic, the *conclusions* could be structured in line with the well-known agent architecture: Beliefs, Desires, Intentions.

Keywords: Nuanced-reasoning logic, Multi-Agent Systems, Sub-symbolic inferences, Stigmergic coordination, Synergy

1 Introduction. From Chrysippus, via Moisil, to Agent Logics

For over 40 years, *determinism and bivalence* of Chrysippean logic were the pillars of Computer Science; likewise, algorithms were the backbone of computer programs, complying with their etymon: *pro-gramma = what is written in advance*. They sufficed for both FORTRAN-like number crunching and COBOL-like business data processing. When early real-time applications (firstly, operating systems) required less autistic programs, algorithms tried to adapt and bizarre terms, such as "unsolicited input", were coined to fit the incipient non-determinism due to user free will. Bivalence not only survived, but also grew in importance strongly backed by hardware. Indeed, in the early 70's, the role of bivalent logic transcended the borders of narrow data processing, penetrating "Computer-Aided x", where x stays for almost any intellectual activity. Thus, "algorithmic reasoning", instead of being perceived as a side effect of "analogue humans loosing the battle with digital computers", became a paradigm in the very sense of Kuhn.

Emerging within this "digital Zeitgeist", nuanced-reasoning [12] was too anti-paradigmatic to redress the balance - at least in IT (besides, it was technologically useless, as most fascinating heresies). Only after the "PC-Windows-WWW" revolution was this "nuanced" kind of fuzzy logic - developed by Zadeh as "computing with words" - acknowledged as an alternative approach to software development (albeit seldom necessary).

On the other hand, after a decade of success stories, within artificial intelligence (AI) - the perpetual stronghold of applied logics and symbolic processing -, expert systems (based on the Newell-Simon hypothesis) began to disappoint, because of their brittleness (in all nuances of the word), showing the actual limits of the symbolic paradigm. The reaction was prompt, overwhelming, and exaggerated: "GOFAI" (*G*ood *O*ld-*F*ashioned *AI*) has to be replaced by "BIC" (*B*iologically *I*nspired *C*omputing), based on sub-symbolic paradigms. The most nihilist and powerful one, i.e. the ethological paradigm (based on the physical-grounding hypothesis), is, for good reasons, still in vogue. However, paradoxically, new, "much nuanced" logics are already used in modern software engineering, tending to become necessary mainly for non-trivial MAS, despite many, major, and obvious adversities.

The paper aims to: a) defend not just those logics but also the inexorable need of symbolic processing, even in systems where intelligent behaviour emerges sub-symbolically (because of its synergistic potential); b) after explaining *why* synergy, show *how* it can be reached. (That is why the title contains the unusual term "(Sub-)Symbolic".) Thus, after a short *history* (Section 2), the *approach* is rendered from an anthropocentric perspective: the agent shall behave naturally (i.e., closer to human behaviour), not the opposite (Section 3). Through this sieve, Section 4 outlines the features and symbolic mechanisms asked for by the paradigm of "computing as intelligent interaction", based on *"nuances of nuanced-reasoning"*. To keep the approach credible, Section 5 sums up recent research showing how *quantifiable interparadigmatic synergy* can be reached - even in advanced challenging domains, such as *stigmergic coordination* - by injecting symbolic reasoning in systems based on sub-symbolic "emergent synthesis". Since for future work the preferred logics are doxastic, the *conclusions* (Section 6) - far from being apodictic - can be structured in line with the well-known agent architecture: Beliefs, Desires, Intentions.

2 History. In Search of Synergy

The research roots are in over 20 papers/articles published in 1997-2002 and synthesised in [3]. After 2002 there are two history strands having the common denominator "looking for synergy in the world of humans and agents": ★ *Stigmergic Coordination*. After minor improvements in 2003, in [13] some (less quantifiable) synergy was achieved deviating from the biological model applied in the Elitist Ant Systems by adding symbolic processing components (firstly adapting the environment and secondly instituting limited central coordination). In [7] a refined experimental model attested that in operational research, through "stigsynergy" the same solution quality could be reached with fewer ants than used in common benchmarks, saving thus at least one order of magnitude of processing time. ★ *Human-Agent Communication*. User-avatar interaction was illustrated in medical captology, employing pathematic agents as virtual therapists [4]. The framework was widened (in the context of broadband communication) to any anthropocentric interface in [5], focusing on the languages enabled by modern multimodal interfaces. On a more abstract level, [6] showed how trans-disciplinary metaphors, applied in communication procedures, can help humanists and technologists get close.

3 Approach. Towards Natural Behaviour of Artificial Entities

Two perspectives guide the approach: anthropocentric systems, as non-negotiable goal, and agent-oriented software engineering (AOSE), as amendable means - depending on long-range effectiveness. (*Anthropocentrism* means focusing on the human being as user, beneficiary, and, ultimately, *raison d'être* of any application or, more general, technology [5]. Here, *"anthropocentric"* is synonymous to *"human-centred"*). The *premises* are:

- *A. Regarding the goal:* ★ Despite their fast rising technological level, most IT applications involving intense human-computer interaction (HCI) have low degree of user acceptance, ignoring the very slogan: "computing as interaction" [1]. ★ That drawback holds mainly for AI systems, widening the gap between humanists and technologists. ★ The main cause: system development is rather *techno*centric than *anthropo*centric. ★ The main neglected human features are: 1★) Invariants: humans are intrinsically analogue in information processing and multimodal in perception. 2★) Prevalent in HCI: humans prefer symbolic communication but sub-symbolic response.

- *B. Regarding the means:* ★ The IT infrastructure is sufficiently advanced (in both facts and trends: nanoelectronics, broadband communication, semantic web, multimodal interfaces, etc.) to allow anthropocentrism for most IT applications. ★ Intelligent system behaviour - whatever that could mean - becomes a crucial user expectation. Regrettably, in AI neither technology, nor design philosophy were yet able to offer it in a user-relevant manner. ★ Nevertheless, agent technology, as AI flagship, proved to be a significant step towards user acceptance. ★ AOSE is not bounded to AI, but tends to become the dominant IT development paradigm [11], [15].

While the first premises in each category are generally accepted, the last ones are debatable (e.g., A4b is rather an "author thesis" and B4 is strongly contested by object-oriented designers). The *corollaries* relevant for the paper are:

- C1. The geometrically increasing computing power (due to Moore's law) promotes at least five factors tending to reduce radically the role of any species of logic in IT - at least for applications affordable on usual configurations: 1★) Since deterministic applications are vanishing, the conventional algorithm is not anymore *program backbone*. 2★) Even when still useful, the conventional algorithm is not anymore the main *programming instrument* (being hidden in procedures easily reached in a host of libraries or being generated by 4GL). 3★) In AI the *symbolic* paradigm is steadily replaced by several *sub-symbolic* ones, based on fine-grain parallelism. 4★) Even when symbols are used, they are *stored* in and *retrieved* from huge and cheap memory, rather than *processed* through sophisticated reasoning schemes (case-based reasoning is just a blatant example). 5★) Cognitive *complexity* of new, sophisticated logics is too high for a designer, when "cut and try" is affordable.

- C2. The rules for human-agent interaction can and should be set by users (at least while we have the Demiurgic privilege of shaping agents as we like it!): 1★) Since *interaction* is carried out through the *interface*, anything behind it is user-irrelevant. 2★) Since *natural* and artificial *intelligence* encounter at interface level, they shall *join*, not *collide*. 3★) To join closer to human demeanour, users should engage

interface agents as *naturally* as possible. 4★) Hence, let agents *behave* more and more *naturally* (e.g., it is not difficult to go beyond gestures to show emotivity, since not *emotion* has to be replicated, but its *appearance* - firstly forged, later more genuine [5]). 5★) Since *interaction* involves *communication*, the *communication procedures* (the term "procedures" is here a prudent, albeit partial, place holder for "language" or, even, "empathy") must be those humans are familiar with (e.g., body language can and shall be added to verbal messages). 6★) Since beside *how* to communicate (the vehicle), it is vital *what* (the message), beyond the *procedures*, there must be a *representational compatibility* between humans and agents (expressed through common ontologies, primitive surrogate of a yet impossible common "Weltanschauung").

If regarding C2.1-C2.4, the blend "symbolic/sub-symbolic" is unclear, C2.5 implies symbols, whereas C2.6 is stronger, implying symbolic inferences. At least some of them shall be based on logic(s).
For short, acknowledging the decline of logic (because of C1), its necessity is asserted (in line with C2). Anyway, the role of desirable features of new logics could be credibly defended - outside large-scale systems, were the proof is futile - only comparing diverse implemented MAS designed with or without employing such logics. Because of C1.5, this is impractical. To weaken this main approach drawback, the argument is split to render two complementary paths, both based on the idea that the blend "symbolic/sub-symbolic" yields synergy: a) axiological perspective: why and what symbolic processing (Section 4, closer to a position paper); b) praxiological perspective: how can symbolic processing be added in experimental sub-symbolic models (Section 5, closer to a technical report).

4 Nuances of Nuanced-Reasoning in Human-Agent Dialogue

It would be both arrogant and absurd if authors lacking educational background in both mathematics and logic would utter value judgments in these fields. Hence: ★ Without claiming that Moisil actually attached to *"nuanced"* other connotation than *"fuzzy"*, bearing in mind his gifted baroque way of catalysing brainstorming, it is legitimate to use undertones of three (partial) synonyms - *"degree"*, *"gradation (sequence, development)"*, *"fine distinction"* - as metaphor sources. ★ All assertions about existing or desirable logics mirror the angle of potential users of such logics, mainly in interface agents and MAS based on stigmergy. They convey "calls for help", not requests, and are uttered as desires. ★ Since, as regards logics dealing with agent-related aspects, for many basic AOSE requests, Fisher's logic [9] seems for a non-specialist by far to be the most responsive and appropriate, all desiderata below refer to it.

■ **Diversified inferences.** Smith's propositional-representation theory should be: a) revisited and thoroughly extended; it shall include all main mechanisms (symbolic or not) employed by humans to infer and to make decisions (even "right-hemisphere based" processes, as educated guess, intuition or gambling); b) applied, depending on the sub-field; such mechanisms should be replicated - as "omomorph", as adequate (not as possible!) - in *agent decision making schemata*. If all of them would reach the elegance and dependability of logic, it would be nice, but let it be yet a kind of "princess lointaine", because in real-world systems most concepts involved tend to become blurred. For instance, even metalogic is now nuanced: *soundness* remains crucial (still - apart from time-critical applications - it can be circumvented through revisable reasoning); *completeness* is more negotiable (the oversimplified solution: "otherwise, nothing happens").
■ **"More time for agents".** Nowadays, any software piece unable to interact efficiently with unpredictable environments (humans included) and with its peers is hardly useful outside toy-problems. That means: parallelism, temporal dimension, non-determinism, reactivity. Corollary: any such program entity has to be implemented as *execution thread* (atomic, sequential, asynchronous and dynamic) [3]. To develop into an agent, the thread needs also non-trivial *informational* and *motivational* components. (However, the "dynamic component" is a treble confusing term: a) it is not a component but the very agent nature; b) the "sense of time" refers to much more than activity - e.g., "waiting" is rather inactivity; c) "dynamic activity" sounds pleonastic from any stance.).
■ **No "start" and no "synchronous agents".** If for e-commerce, it is conceivable to consider that the entire world restarts with each transaction, for process control (even for discrete manufacturing) such eternal re-birth is practically excluded. Moreover, it is against the very spirit of: a) the (still dominant) "client-server" paradigm (the tailor is not spawned every time a client needs new clothes); b) real-time software engineering (to react timely to environment stimuli, the thread must exist to handle the interrupt); c) agency itself: the basic feature of autonomy (implying asynchronous behaviour) is endangered. Luckily, current timers permit a "fine-grain universal metronome", avoiding the costly implication: "asynchronously executing agents → temporal logic of the reals".

Thus, "asynchronously executing agents" should be perceived as pleonasm, despite their logic is still based upon a discrete model of time, with both infinite past and future. (In real-world MAS, there is no "Big-Bang".)

■ **No "negative introspection".** Unable to comment upon the advantages of ideal doxastic logics outside large-scale MAS, the authors feel that positive introspection is highly desirable but that assuming the negative one is ineffective for both agents and humans. Thus, if it makes sense and simplifies the features, maybe **KD4**, not **KD45**.

■ **No more certitude. Less checking.** Until agent logics offer mechanisms to deal with uncertainty, at least, in simple expressions, the "ugly chasm" separating formal theory and practical system development [9] cannot be avoided. Just a plain example of a badly needed such mechanism: exception handling. Even primeval animals move "algorithmically" ("*if* gap *then* get round, *else* go on") only a few steps, in very hostile environments. Moreover, reaction to stimuli cannot mean perpetual looking for the stimulus. (Instead, the stimulus causes an interrupt that can be treated as exception.) The cardinal hindrance stems not from logic, but from the mechanisms employed: neither nature, nor technology can afford in the long run mechanisms involving large amount of testing because they are too time-consuming tools: "*if* temperature $>$ n ^{o}C *then* alarm". Thus, the main problem is not the semantics of *"unless"*, but the repeated checking of *"if"*. From this angle, the semantics of *"unless"* in Reiter's default logic would be more tempting if it would be rather diachronic than synchronic (a bird *is* or *is not* a penguin but will never *become* one). However, a kind of *M* operator meaning roughly "while no alarm is heard it is consistent to believe that nothing happened". Indeed, the agent is condemned to be a risk-taker, *hearing* (reactively) the environment, not *listening* (proactively) to it: the agent stops performing a task only if he hears the alarm bell. The point is that this *"if"* belongs to the metalanguage and does not involve thermometer reading! Perhaps a non-monotonic logic with "Reiter-*unless*" inserted in a temporal logic with "Fisher- *unless*" is what designers dream of. (Since dreams are forward-thinking, maybe more: a graphical "flowchart-like" symbol of this *M* shall be understood by an interpreter of an "AOSE-ML" -without "object legacy" - that can create code for defining, raising, propagating, and handling exceptions.

5 Down to Ants: Synergy, Stigmergy, AND Symbols

Since as regards stigmergic coordination the research was recently summarised in [7], [8], [13] and the current results are presented in [7], here, only the approach and some relevant aspects of achieving synergy through grafting symbolic processing onto sub-symbolic systems are emphasised. The AND written in capitals emphasises the similarity with the synonymous boolean operator, i.e. synergy is searched for in all possible combinations.

The MAS that relies on sub-symbolic processing more than any other is the biologically inspired *Ant System* (AS) where the sub-symbolic echelon is represented by the *pheromones* in such a way that global information is available locally. Moreover, this system is not only sub-symbolic by itself but it also manifests *autopoiesis* (it emerges subsymbolic) and the trouble to understand what is in fact going on at system level, is less upsetting than in the case of more familiar sub-symbolic paradigms (as artificial neural networks or evolutionary algorithms) since ant behaviour is easier to follow due to its simplicity.

The stigmergy related to MAS, "describes a form of asynchronous interaction and information exchange between agents mediated by an 'active' environment", or "the production of certain behaviour in agents as a consequence of the effects produced in the local environment by previous behaviour". In this context: "the agents are simple, reactive, and unaware of other agents or of the emerging complex activities of the agent society; the environment is an important mechanism to guide activities of these agents and to accumulate information about ongoing activities of the whole agent society" [13].

Whereas in [2], [10], [14], the approach was mainly based on self-organization, the approach is an alternative one by obtaining synergy through adding symbolic processing (firstly adapting the environment and secondly instituting limited central coordination). As shown in [7], the AS manifests a threshold and it depends on problem type and complexity; the same solution quality can be obtained with fewer ants than used in common benchmarks, saving thus at least one order of magnitude of processing time.

Details can be found in [13] (improvements to conventional EAS), [8] (motivation, approach and new perspective), and [7] (experimental results about moving the threshold - in fact modifying the sigmoid function to improve efficiency). Possible scientific openings - e.g. whether in real-life problems there are instances when "many starts from four" - can be found also in [7].

6 Conclusions: Beliefs, Desires, Intentions

The conclusions are presented within the BDI frame not just to keep up the atmosphere, but because: a) the conclusions are far from being apodictic and the logics preferred for MAS are doxastic; b) using the meanings given by Smets, the belief functions have rather dispersed values, and the plausibility functions have quite low values for Section 4 and some assertions of Section 3; c) the largest part of Section 3 is actually a gathering of desires; d) intentions is more humble than "future work"; d) if we intend to interact keener with agents, we have to make steps towards common ontologies - preferably based on success stories.

■ ***Beliefs:*** ★ Despite the fall of conventional algorithms and the fast rise of sub-symbolic paradigms, symbolic processing is unavoidable in AOSE and agent logics become necessary even outside large-scale systems. ★ An essential problem in designing agents is implementing their *reactivity*; main cause: current development environments admit rather very poor exception handling. ★ Even MAS based on the most radical sub-symbolic paradigm (stigmergy being "a-symbolic" par excellence), become more effective grafting upon symbolic processing. ★ Taking into account the increasing weight of MAS acting as man-machine systems, the anthropocentric perspective requires that human-agent communication should be the model for agent-agent communication. ★ Although the brains-surrogate of current agents is still primitive, it shall have two hemispheres, as human do. The left hemisphere, where logic is king, is designed predominantly to implement pro-activeness, whereas the right one, as realm of its instincts, emerges sub-symbolically, and is the main source of reactivity (again, similar to humans).

■ ***Desires:*** *They are addressed to future agent logics, from an outsider (but outspoken AOSE) perspective:* ★ Tackle neglected problems common to all kinds of agent-based systems (dwarfs and trolls welcomed). ★ Give us sectorial solutions. They are just fine to begin with. Completeness - in its polysemy - can follow. (If the MAS is sound, nobody minds if agents manifest a bit of schizophrenia.) ★ Don't give us sectorial approaches. They are less applicable (e.g., time without uncertainty or vice versa). ★ Let MAS be lasting, even if some agents are mortal. ★ Don't condemn MAS to act synchronously. Both environment and users are too capricious to accept it. (Instead, we promise to be happy with discrete time.) ★ Don't sentence us to perpetual testing. To rephrase Dijkstra: (the condition in) *if* is harmful. (Allow us to handle exceptions, and we promise not to exaggerate eliminating all "iffs".) ★ Help us pass the mental Rubicon separating objects from agents. (No agent is fond of being considered "intelligent and responsive like an object".)

■ ***Intentions:*** ★ As regards stimergic coordination, the intentions are those states in [7]: for short, increasing "stigsynergy". ★ Showing how agent reactivity can be significantly improved, through exception-driven multimodal interfaces. ★ Trying dialectics as inference mechanism for negotiation strategies used by e-commerce agents.

Acknowledgment. This paper is related to the EU-sponsored action COST-298.

References

[1] "AgentLink Roadmap: Overview and Consultation Report", *AgentLink III. Agent based computing*, University of Southampton, 2004.

[2] Banzhaf, W., "Self-organizing Systems", *Encyclopedia of Physical Science and Technology,* Academic Press, New York, **14**, pp. 589-598, 2002.

[3] Bărbat, B.E., "Sisteme inteligente orientate spre agent", *Ed. Academiei Române*, Bucureşti, 467 pages, 2002.

[4] Bărbat, B.E., "The Avatar - A Pseudo-Human Extending the Genuine One", *The good, the bad and the irrelevant: The user and the future of information and communication technologies,* (L. Haddon et al, Eds.), pp. 38-42, Media Lab/University of Art and Design, Helsinki, 2003.

[5] Bărbat, B.E., "The Impact of Broad-Band Communication upon HMI Language(s). (Chapter 7.) Communicating in the world of humans and ICTs. (Chapter 8.)", *COST Action 269. e-Citizens in the Arena of Social and Political Communication*, (L. Fortunati, Ed.), pp. 113-142, EUR21803, Office for Official Publications of the European Communities, Luxembourg, 2005.

[6] Bărbat, B.E., N. Bulz, "E-World and Real World. From Analysing Concepts towards Integrating Approaches", *Proc. of the WOSC 13th International Congress of Cybernetics and Systems,* University of Maribor in cooperation with Encyclopaedia of Life Support Systems, **1**, pp. 47-57, 2005.

[7] Bărbat, B.E., S.C. Negulescu, C.B. Zamfirescu, "Human-Driven Stigmergic Control. Moving the Threshold", *Proc. of the 17th IMACS World Congress (Scientific Computation, Applied Mathematics and Simulation)*, Paris, July 11-15, 2005.

[8] Bărbat, B.E., C.B. Zamfirescu, S.C. Negulescu, "The Best from Ants and Humans: Synergy in Agent-Based Systems", *Studies in Informatics and Control Journal*, **13**, 1, 47-59, 2004.

[9] Fisher, M., "Temporal Development Methods for Agent-Based Systems", *Autonomous Agents and Multi-Agent Systems*, **10**, pp. 41-66, Springer Science + Business Media Inc., 2005.

[10] Knyazeva, H., Haken, H., "Synergetics of Human Creativity. Dynamics, Synergetics, Autonomous Agents", *ANonlinear Systems Approaches to Cognitive Psychology and Cognitive Science Singapore: World Scientific*, pp. 64-79, 1999.

[11] Luck, M., McBurney, P., Priest, C., "A Manifesto for Agent Technology: Towards Next Generation Computing", *Autonomous Agents and Multi-Agent Systems*, **9**, 203-252, Kluwer Academic Publishers, 2004.

[12] Moisil, Gr.C., "Lecţii despre logica raţionamentului nuanţat", *Ed. Ştiinţifică şi enciclopedică*, Bucureşti, 1975.

[13] Negulescu, S.C., Bărbat, B.E., "Enhancing the Effectiveness of Simple Multi-Agent Systems through Stigmergic Coordination", *Fourth International ICSC Symp. on ENGINEERING OF INTELLIGENT SYSTEMS (EIS 2004)*, ICSC-NAISO Academic Press Canada, 149 (Abstract; full paper on CD-ROM enclosed), 2004.

[14] Parunak, H.V.D., Brueckner, S., John A. Sauter, Matthews, R., "Global Convergence of Local Agent Behavior", *Submitted to The Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05)*, 2005, Available at http://www.erim.org/ vparunak/ AAMAS05Converge.pdf.

[15] Zambonelli, F., Omicini, A., "Challenges and Research Directions in Agent-Oriented Software Engineering", *Autonomous Agents and Multi-Agent Systems*, **9**, 253-283, Kluwer Academic Publishers, 2004.

Boldur E. Bărbat
"Lucian Blaga" University of Sibiu
Faculty of Science
Address: 5-7, Ion Raţiu Street, Sibiu, 550012, România
E-mail: bbarbat@gmail.com

Sorin C. Negulescu
"Lucian Blaga" University of Sibiu
Faculty of Engineering
Address: 4, Emil Cioran Street, Sibiu, 550025, România
E-mail: sorin_negulescu@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 74-79

Optimal Piecewise Smooth Interpolation of Experimental Data

Alexandru Mihai Bica, Mircea Curilă, Sorin Curilă

Abstract: The notions of optimal piecewise smooth interpolation and oscillation of interpolation type are introduced. This notions are used and illustrated on some cubic piecewise smooth interpolation procedures and on some cubic splines. It is obtained a cubic spline of interpolation having minimal quadratic oscillation in average, as an application of the least squares method.

Keywords: piecewise smooth interpolation, optimal smooth approximations, oscillation of interpolation type, quadratic oscillation in average.

1 Introduction

Definition 1. Let $\Delta \in Div[a,b]$,

$$\Delta : a = x_0 < x_1 < ... < x_{n-1} < x_n = b,$$

and $y = (y_0, y_1, ..., y_n) \in \mathbb{R}^{n+1}$. Let $L(f) : [a,b] \longrightarrow \mathbb{R}$, a function which interpolate the points $(x_i, y_i),\ i = \overline{0,n}$, that is, $L(f)(x_i) = y_i,\ \forall i = \overline{0,n}$. We say that the interpolation realized by $L(f)$ is optimal if there exist $k \in \mathbb{N}$ and a functional $J_{y,\Delta} : C^k[a,b] \longrightarrow \mathbb{R}$, such that for given $Y \subseteq C^k[a,b]$ we have,

$$J_{y,\Delta}(L(f)) = \min\{J_{y,\Delta}(g) : g \in Y, \quad g(x_i) = y_i,\ \forall i = \overline{0,n}\}.$$

Such optimal cubic splines.can be found in [7], where $Y = C^2[a,b]$ and $J_{y,\Delta}(g) = \int_a^b g''(x)\,dx$.

Now, we present another way to obtain optimal smooth interpolation. The notions of oscillation of interpolation type and quadratic oscillation in average, introduced here, is proper for any interpolation functions. Consider an interval $[a,b]$ and a division $\Delta_n \in Div[a,b]$ of this interval,

$$\Delta_n : a = x_0 < x_1 < ... < x_{n-1} < x_n = b. \tag{1}$$

Let $h_i = x_i - x_{i-1}$ and $I_i = [x_{i-1}, x_i]$, $\forall i = \overline{1,n}$. Let $y_i \in \mathbb{R}$, $i = \overline{0,n}$ and $y = (y_0, ..., y_n)$. For each $i = \overline{1,n}$ we define $D_i : I_i \longrightarrow \mathbb{R}$, by

$$D_i(x) = y_{i-1} + \frac{y_i - y_{i-1}}{h_i} \cdot (x - x_{i-1}),\ \forall x \in I_i. \tag{2}$$

The graph of D_i is the line joining the points (x_{i-1}, y_{i-1}) and (x_i, y_i).

We construct $D(y) : [a,b] \longrightarrow \mathbb{R}$, such that $D(y)(x) = D_i(x), \quad \forall x \in [x_{i-1}, x_i],\ \forall i = \overline{1,n}$. It is easy to see that the graph of $D(y)$ is the polygonal line interpolating the points $(x_i, y_i),\ i = \overline{0,n}$. For any $i = \overline{1,n}$, let $\overline{D_i}(y) : [a,b] \longrightarrow \mathbb{R}$, defined by

$$\overline{D_i}(y)(x) = \begin{cases} 0, & x < x_{i-1} \\ D_i(x), & x \in [x_{i-1}, x_i] \\ 0, & x > x_i. \end{cases} \tag{3}$$

Let $f : [a,b] \longrightarrow \mathbb{R}$, continuous such that $f(x_i) = y_i,\ \ i = \overline{0,n}$ and denote by f_i, the restriction of f to the interval $[x_{i-1}, x_i]$ for any $i = \overline{1,n}$. For any $i = \overline{1,n}$ we define $\overline{f_i} : [a,b] \longrightarrow \mathbb{R}$, by

$$\overline{f_i}(x) = \begin{cases} 0, & x < x_{i-1} \\ f_i(x), & x \in [x_{i-1}, x_i] \\ 0, & x > x_i. \end{cases} \tag{4}$$

Definition 2. A function $J_{y,\Delta} : C[a,b] \longrightarrow \mathbb{R}$, is oscillation of interpolation type corresponding to the division $\Delta \in Div[a,b]$ and to $y = (y_0, y_1, ..., y_n) \in \mathbb{R}^{n+1}$, if for any $f \in C[a,b]$ having $f(x_i) = y_i,\ \forall i = \overline{0,n}$, the following properties holds :
(i) (positivity) : $J_{y,\Delta}(f) \geq 0, \quad \forall f \in C[a,b]$ and

$$J_{y,\Delta}(f) = 0 \Longleftrightarrow f = D(y),$$

(ii) (absolute homogeneity)

$$J_{y,\Delta}(\alpha \cdot f) = |\alpha| \cdot J_{y,\Delta}(f), \quad \forall \alpha \in \mathbb{R}^*, \quad \forall f \in C[a,b]$$

(iii) (monotony) : the following implication is true :

$$\left\|\overline{f_i} - \overline{D_i}\right\|_C \leq \left\|\overline{g_i} - \overline{D_i}\right\|_C \quad , \quad \forall i = \overline{1,n} \Longrightarrow J_{y,\Delta}(f) \leq J_{y,\Delta}(g), \quad \forall f \in C[a,b],$$

where, $\|\cdot\|_C$ is the Chebyshev's norm for bounded functions.

Definition 3. (see [2])The quadratic oscillation in average corresponding to Δ_n and $y = (y_0, y_1, ..., y_n)$, is the function

$$\rho(f;\Delta_n,y) : C[a,b] \longrightarrow \mathbb{R}, \tag{5}$$

defined by

$$\rho(f;\Delta_n,y) = \sqrt{\int_a^b (\sum_{i=1}^n [\overline{f_i}(x) - \overline{D_i}(y)(x)]^2) dx}, \tag{6}$$

where, $\overline{f_i}$ and $\overline{D_i}(y)$ are as above.

Remark 1. We see that $\rho(f;\Delta_n,y) \geq 0$, $\forall f \in C[a,b]$ with $f(x_i) = y_i$, $\forall i = \overline{0,n}$ and $\rho(f;\Delta_n,y) = 0 \Longleftrightarrow f = D(y)$. Moreover,

$$\rho(\alpha \cdot f;\Delta_n,\alpha \cdot y) = |\alpha| \cdot \rho(f;\Delta_n,y), \ \forall \alpha \in \mathbb{R}^*, \quad \forall f \in C[a,b]$$

and

$$\left\|\overline{f_i} - \overline{D_i}\right\|_C \leq \left\|\overline{g_i} - \overline{D_i}\right\|_C \quad , \quad \forall i = \overline{1,n} \Longrightarrow \rho(f;\Delta_n,y) \leq \rho(g;\Delta_n,y).$$

Therefore, the quadratic oscillation in average is a oscillation of interpolation type corresponding to the division $\Delta_n \in Div[a,b]$.and to $y = (y_0, y_1, ..., y_n)$. Since,

$$\rho(f;\Delta_n,y) \leq \sqrt{n(b-a)} \cdot \|f - D\|_C$$

we infer that minimizing the quadratic oscillation in average, we will minimize the distance between the function f and the polygonal line. Consequently, will be minimized the oscillations of f on $[a,b]$.

Since,

$$[\overline{f_i}(x) - \overline{D_i}(x)]^2 = \begin{cases} 0, & x < x_{i-1} \\ [\overline{f_i}(x) - \overline{D_i}(x)]^2, & x \in [x_{i-1}, x_i] \\ 0, & x > x_i. \end{cases}, \ \forall i = \overline{1,n}$$

we infer that the function given on $[a,b]$ by the sum, $\sum_{i=1}^n [\overline{f_i}(x) - \overline{D_i}(x)]^2$ is Riemann integrable on $[a,b]$.

Remark 2. We can attach the notion of quadratic oscillation in average to any interpolation functions : Lagrange polynomial, Hermite polynomial, Birkhoff lacunary interpolation polynomial (see [11]), spline function of interpolation with any degree ≥ 2 (see [7], [8] and [10]), piecewise Hermite polynomial interpolation (see [1], [7] and [9]) and others.

2 Examples of optimal piecewise smooth interpolation

1. In [7], for the cubic spline generated by two point boundary conditions, $s : [a,b] \longrightarrow \mathbb{R}$, with $s''(a) = s''(b) = 0$, we have $Y = C^2[a,b]$ and $J_{y,\Delta} : C^2[a,b] \longrightarrow \mathbb{R}$, $J_{y,\Delta}(g) = \int_a^b g''(x)\,dx$. It is proved that s realize optimal smooth interpolation of the data (x_i, y_i), $i = \overline{0,n}$, since,

$$J_{y,\Delta}(s) = \int_a^b s''(x)\,dx = \min\{J_{y,\Delta}(g) : g \in C^2[a,b], \quad g(x_i) = y_i, \ \forall i = \overline{0,n}\}.$$

Moreover, in [2] it is obtained the cubic spline generated by initial conditions, which realize optimal smooth interpolation in the same way as above, for the same functional $J_{y,\Delta}$ and $Y = C^2[a,b]$.

2. The quadratic oscillation in average is used in [2] to obtain the cubic spline $s : [a,b] \longrightarrow \mathbb{R}$, having minimal quadratic oscillation in average and the restriction to each subinterval $[x_{i-1},x_i]$, $i = \overline{1,n}$ as :

$$s_i(x) = \frac{1}{6h_i}(M_i - M_{i-1})(x-x_{i-1})^3 + \frac{M_i}{2}(x-x_{i-1})^2 + m_i(x-x_{i-1}) + y_{i-1}, \tag{7}$$

$\forall x \in [x_{i-1},x_i]$, where $h_i = x_i - x_{i-1} \quad \forall i = \overline{1,n}$. The functions s_i, $i = \overline{1,n}$, are determined solving the initial value problems :

$$\begin{cases} s_i''(x) = M_i + \frac{1}{h_i}(M_i - M_{i-1})(x-x_{i-1}) \\ s_i(x_{i-1}) = y_{i-1} \\ s_i'(x_{i-1}) = m_{i-1}, \end{cases} \tag{8}$$

where $M_i = s_i''(x_i)$. Since $s \in C^2[a,b]$, the conditions $s(x_i) = y_i$, $s'(x_i) = m_i$, $i = \overline{1,n}$, lead to the relations :

$$\begin{cases} M_i = \frac{6}{h_i^2} \cdot (y_i - y_{i-1}) - \frac{6m_{i-1}}{h_i} - 2M_{i-1} \\ m_i = \frac{3}{h_i} \cdot (y_i - y_{i-1}) - 2m_{i-1} - \frac{M_{i-1}h_i}{2} \end{cases}, \; i = \overline{1,n}. \tag{9}$$

From these relations we infer that m_i, M_i, $i = \overline{1,n}$, are uniquely determined starting by $y_0, y_1, ..., y_n$, m_0 and M_0. Then, in [8] it is proved that the cubic spline is uniquely determined by $y_0, y_1, ..., y_n$, m_0 and M_0. In [2] are determined the values m_0 and M_0 such that s to have minimal quadratic oscillation in average, using the least squares method. Here, in the sense of Definition 1, the set $Y \subseteq C^2[a,b]$ is,

$$Y = \{g \in C^2[a,b] : g_i = g\,|_{[x_{i-1},x_i]},\; g_i(x) = A_i(x) \cdot u_i + B_i(x) \cdot v_i + (x-x_{i-1}) \cdot w_i + y_{i-1},\;\; i = \overline{1,n}\}$$

for given y_{i-1},

$$A_i(x) = -\frac{1}{6h_i} \cdot (x-x_{i-1})^3, \quad B_i(x) = \frac{1}{6h_i} \cdot (x-x_{i-1})^3 + \frac{1}{2} \cdot (x-x_{i-1})^2,$$

and the functional is $J_{y,\Delta_n} : C^2[a,b] \longrightarrow \mathbb{R}$,

$$J_{y,\Delta_n}(g) = \int_a^b (\sum_{i=1}^n [\overline{f_i}(x) - \overline{D_i}(y)(x)]^2)dx,$$

where, $\overline{D_i}(y)$ and $\overline{f_i}$ are as in (3) and (4).

3. Other use of the quadratic oscillation in average to obtain optimal cubic piecewise smooth interpolation can be found in [3] and [4]. Analogous, in [3] are determined y_i', $i = 0$, $i = 1$, $i = \overline{n-2,n}$ for which the Akima's method of piecewise smooth interpolation became optimal for the interpolation function $F : [a,b] \longrightarrow \mathbb{R}$, having the restrictions to the intervals $[x_{i-1},x_i]$, $\quad i = \overline{1,n}$,

$$F_j(x) = \frac{(x_i-x)^2 \cdot (x-x_{i-1})}{h_i^2} \cdot y_{i-1}' - \frac{(x-x_{i-1})^2 \cdot (x_i-x)}{h_i^2} \cdot y_i' +$$

$$+ \frac{(x_i-x)^2 \cdot [2(x-x_{i-1}) + h_i]}{h_i^3} \cdot y_{i-1} + \frac{(x-x_{i-1})^2 \cdot [2(x_i-x) + h_i]}{h_i^3} \cdot y_i,$$

where, $h_i = x_i - x_{i-1}$. Here, y_i', $i = \overline{2,n-3}$ are calculated by the Akima's procedure (see [1] and [7]). In [4], using the least squares method, is obtained the piecewise smooth interpolation function (constructed for the first time in [9]) with minimal quadratic oscillation in average. In [3] and [4], the functional $J_{y,\Delta}$ is defined on $C^1[a,b]$ and minimized on different subsets $Y \subset C^1[a,b]$.

3 Main result

Let $\Delta \in Div[a,b]$,

$$\Delta : a = x_0 < x_1 < ... < x_{n-1} < x_n = b$$

and $y=(y_0,...,y_n)\in\mathbb{R}^{n+1}$. Let $h_i=x_i-x_{i-1},\quad i=\overline{1,n}$. Consider the cubic spline from [7], $s:[a,b]\longrightarrow\mathbb{R}$, $s\in C^2[a,b]$, interpolating the points $(x_i,y_i),\ i=\overline{0,n}$, and having the restrictions $s_i,\ i=\overline{1,n}$, to the subintervals $[x_{i-1},x_i]$ of the division Δ,

$$s_i(x)=\left[\frac{(x-x_{i-1})^3}{6h_i}-\frac{h_i(x-x_{i-1})}{6}\right]\cdot M_i+\left[\frac{(x_i-x)^3}{6h_i}-\frac{h_i(x_i-x)}{6}\right]\cdot M_{i-1}+$$

$$+\frac{x_i-x}{h_i}\cdot y_{i-1}+\frac{x-x_{i-1}}{h_i}\cdot y_i \overset{notation}{=} A_i(x)\cdot M_i+B_i(x)\cdot M_{i-1}+$$

$$+C_i(x)\cdot y_{i-1}+E_i(x)\cdot y_i,\qquad \forall x\in[x_{i-1},x_i],\quad \forall i=\overline{1,n}.$$

Let $D_i,\quad i=\overline{1,n}$ as in (2). We can see that we have,

$$\int_a^b(\sum_{i=1}^n[\overline{s_i}(x)-\overline{D_i}(y)(x)]^2)dx=\sum_{i=1}^n(\int_a^b[\overline{s_i}(x)-\overline{D_i}(y)(x)]^2dx)=\sum_{i=1}^n\int_{x_{i-1}}^{x_i}[s_i(x)-D_i(x)]^2dx.$$

Therefore, the quadratic oscillation in average $\rho(f;\Delta_n,y)$ from Definition 3 can be defined as well as by the formula :

$$\rho(f;\Delta_n,y)=\sqrt{\sum_{i=1}^n\int_{x_{i-1}}^{x_i}[f_i(x)-D_i(x)]^2dx}.$$

Since $s\in C^1[a,b]$ and

$$s_i'(x)=\frac{M_i\cdot(x-x_{i-1})^2-M_{i-1}\cdot(x_i-x)^2}{2h_i}+\frac{y_i-y_{i-1}}{h_i}-\frac{h_i(M_i-M_{i-1})}{6}$$

$\forall x\in[x_{i-1},x_i],\quad \forall i=\overline{1,n}$, the smoothness conditions $s_i'(x_i)=s_{i+1}'(x_i),\quad \forall i=\overline{1,n-1}$ lead to the $n-1$ relations,

$$\frac{h_iM_{i-1}}{6}+\frac{h_i+h_{i+1}}{3}\cdot M_i+\frac{h_{i+1}M_i}{6}=\frac{y_{i+1}-y_i}{h_{i+1}}-\frac{y_i-y_{i-1}}{h_i},\quad \forall i=\overline{1,n-1}. \tag{10}$$

The relations (10) obtained on the interior knots $x_i,\quad i=\overline{1,n-1}$ fix the values $M_i,\quad i=\overline{1,n-1}$. Therefore, we will consider the residual with M_0 and M_n as variables,

$$R(M_0,M_n)=\sum_{i=1}^n\int_{x_{i-1}}^{x_i}[s_i(x)-D_i(x)]^2dx.$$

Theorem 1. There exist unique $M_i,\quad i=\overline{0,n}$ which minimize the quadratic oscillation in average $\rho(s;\Delta_n,y)=\sqrt{\sum_{i=1}^n\int_{x_{i-1}}^{x_i}[s_i(x)-D_i(x)]^2dx}$. Moreover, if $f\in C^4[a,b]$ then the error estimation is :

$$\|f-s\|_C\le[\|f''\|_C+\max\{M_i:i=\overline{0,n}\}]\cdot h^2+m\cdot h,$$

where, $m>0$ is an upper bound of $\{|f'(x_i)-m_i|:i=\overline{0,n}\}$ and $h=\max\{h_i:i=\overline{1,n}\}$. Consequently, s realize an optimal piecewise smooth interpolation.

Proof. By the least squares method we consider the system

$$\frac{\partial R}{\partial M_0}=0,\qquad \frac{\partial R}{\partial M_n}=0$$

that is,

$$\begin{cases}\frac{\partial}{\partial M_0}\int_{x_0}^{x_1}[s_1(x)-D_1(x)]^2dx=0\\ \frac{\partial}{\partial M_n}\int_{x_{n-1}}^{x_n}[s_n(x)-D_n(x)]^2dx=0\end{cases}\iff$$

$$\Longleftrightarrow \begin{cases} M_0 \int\limits_{x_0}^{x_1} [A_1(x)]^2 dx + M_1 \int\limits_{x_0}^{x_1} A_1(x) B_1(x) dx = q_1 \\ M_{n-1} \int\limits_{x_{n-1}}^{x_n} A_n(x) B_n(x) dx + M_n \int\limits_{x_{n-1}}^{x_n} [B_n(x)]^2 dx = q_2 \end{cases}, \tag{11}$$

where,

$$q_1 = -\int\limits_{x_0}^{x_1} A_1(x) \cdot [C_1(x) \cdot y_0 + E_1(x) \cdot y_1 - D_1(x)] dx$$

$$q_2 = -\int\limits_{x_{n-1}}^{x_n} B_n(x) \cdot [C_n(x) \cdot y_{n-1} + E_n(x) \cdot y_n - D_n(x)] dx.$$

The system (11) is, after elementary calculus,

$$\begin{cases} \frac{2}{9} h_1 M_0 + \frac{1}{4} h_1 M_1 = \frac{35}{6h_1} \cdot (y_1 - y_0) \\ \\ \frac{1}{4} h_n M_{n-1} + \frac{2}{9} h_n M_n = \frac{35}{6h_n} \cdot (y_n - y_{n-1}) \end{cases}.$$

Together with the relations (10), we obtain the system,

$$\begin{cases} \frac{2}{9} h_1 M_0 + \frac{1}{4} h_1 M_1 = \frac{35}{6h_1} \cdot (y_1 - y_0) \\ \\ \frac{h_i M_{i-1}}{6} + \frac{h_i + h_{i+1}}{3} \cdot M_i + \frac{h_{i+1} M_i}{6} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}, \quad \forall i = \overline{1, n-1} \\ \\ \frac{1}{4} h_n M_{n-1} + \frac{2}{9} h_n M_n = \frac{35}{6h_n} \cdot (y_n - y_{n-1}) \end{cases}$$

for which it is easy to prove that have unique solution $(M_0, ..., M_n)$. Moreover, the Hessian matrix of $R(M_0, M_n)$ is :

$$\begin{pmatrix} 2 \int\limits_{x_0}^{x_1} [A_1(x)]^2 dx & 0 \\ 0 & 2 \int\limits_{x_{n-1}}^{x_n} [B_n(x)]^2 dx \end{pmatrix}$$

and then the residual $R(M_0, M_n)$ became minimal. These lead to minimal quadratic oscillation in average. Let arbitrary $x \in [a, b]$. Then there exist $i = \overline{1, n}$ such that $x \in [x_{i-1}, x_i]$. Consequently,

$$|f(x) - s(x)| = |f(x) - s(x) - (s(x_i) - f(x_i))| \leq \int\limits_{x_{i-1}}^{x} |f'(t) - s'(t)| dt \leq$$

$$\leq \int\limits_{x_{i-1}}^{x} [|f'(t) - f'(x_{i-1})| + |f'(x_{i-1}) - m_{i-1}| + |s'(x_{i-1}) - s'(t)|] dt.$$

We have,

$$|f'(t) - f'(x_{i-1})| \leq \|f''\|_C \cdot |t - x_{i-1}| \leq \|f''\|_C \cdot h, \quad \forall i = \overline{1, n}.$$

The derivatives $f'(x_{i-1})$ on the knots can be obtained using the classical numerical derivation formulas (see [6]). For instance,

$$|f'(x_0) - y'_0| \leq \frac{1}{6h'} \cdot |-11y_0 + 18y_1 - 9y_2 + 2y_3 - y'_0| + \frac{h''^3}{4} \cdot \left\| f^{(4)} \right\|_C = d_0,$$

where, $h' = \min\{h_i : i = \overline{1,3}\}$ and $h'' = \max\{h_i : i = \overline{1,3}\}$. Similar estimations can be obtained on the other knots,

$$|f'(x_i) - y'_i| \leq d_i, \qquad \forall i = \overline{1, n}.$$

Let $m = \max\{d_i : i = \overline{0, n}\}$. Then, $|f'(x_{i-1}) - m_{i-1}| \leq m, \qquad \forall i = \overline{1, n}$.
On the other hand, since $s \in C^2[a, b]$, we have,

$$|s'(x_{i-1}) - s'(t)| \leq \|s''\|_C \cdot |t - x_{i-1}| \leq \|s''\|_C \cdot h, \quad \forall i = \overline{1, n}.$$

Morover, $\|s''\|_C = \max\{M_i : i = \overline{0,n}\}$. Follows that,

$$\|f-s\|_C = \max\{|f(x)-s(x)| : x \in [a,b]\} \leq \int_{x_{i-1}}^{x} [\left\|f''\right\|_C \cdot h + m + \max\{M_i : i = \overline{0,n}\} \cdot h]dt$$

$$\leq [\left\|f''\right\|_C + \max\{M_i : i = \overline{0,n}\}] \cdot h^2 + m \cdot h.$$

□

Finaly, we can mention that the above techniques can be followed to obtain varies examples of optimal piecewise smooth interpolation.

The present paper is a continuation of the cooperation of the authors in the interpolation procedures applied in image processing, some of the results being published in [5].

References

[1] H. Akima, *A new method for interpolation and smooth curve fitting based on local procedure*, J. of Assoc. for Comp. Machinery, Vol. 1,4, pp. 589-600, 1970.

[2] A. M. Bica, *Mathematical models in biology governed by differential equations*, PhD Thesis, " Babes-Bolyai " University Cluj-Napoca, 2004.

[3] A. M. Bica, *Optimal improvement of the Akima's method of piecewise smooth interpolation*, (submited).

[4] A. M. Bica, *Optimal properties in piecewise smooth interpolation*, (submited).

[5] M. Curilă, S. Curilă, *Digital processing of the images spoiled by air agents*, Oradea University Press, 2004 (in Romanian).

[6] B. Demidovitch, I. Maron, *Elements de calcul numerique*, Ed. Mir Moscou 1987.

[7] C.Iacob (ed.), *Classical and modern mathematics, vol. 4*, Ed. Tehnica, Bucharest 1983 (in Romanian).

[8] C.Iancu, *On the cubic spline of interpolation*, Seminar of Functional Analysis and Numerical Methods, Preprint no.4, pp. 52-71, 1981.

[9] K.Ichida, F.Yoshimoto, T. Kiyono, *Curve Fitting by a Piecewise Cubic Polinomial*, Computing, vol. 16, pp. 329-338, 1976.

[10] G. Micula, S. Micula, *Handbook of splines*, Mathematics and its applications, 462, Kluwer Academic Publishres, 1999.

[11] D. D. Stancu, Gh. Coman, O. Agratini, R. Trîmbiţaş, *Numerical analysis and approximation theory, vol.1*, Presa Universitara Clujeana, 2001 (in Romanian).

Alexandru Mihai Bica
University of Oradea
Department of Mathematics and Informatics
Str. Universitatii no. 1,410087, Oradea, Romania
E-mail: abica@uoradea.ro

Mircea Curilă
University of Oradea
Faculty of Environment Protection

Sorin Curilă
University of Oradea
Faculty of Electrotechnics and Computer Science

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 80-85

Alternative Solutions toward IPv4/IPv6 Multicast

Tudor Mihai Blaga, Virgil Dobrota, Gabriel Lazar, Bogdan Moraru

Abstract: The paper presents one alternative solution for IPv4 multicast: CastGate. This technology provides seamless access to multicast content through auto-tunneling. It is intended as a transition step toward native multicast, that will lead to a increase in the number of multicast users. Alternative technologies offer solutions for problems which are not addressed by the native multicast model. The proposed enhancements refer to adding PIM-SM support and to the possibility of migrating from IPv4 to IPv6 multicast on CastGate architectures.

1 Introduction

IP multicast is a Network Layer mechanism to support applications where data needs to be sent from a source to multiple receivers (point-to-multipoint). The applications based on this concept could be for instance conferencing systems, software updates and on-demand video distribution. A major benefit of using multicast is the considerable decrease of the network and server load. It reduces the number of packets sent when the destination is a group of nodes.

Traditionally streaming media content is offered using unicast. In this case, the bandwidth requirements increase linearly with the number of receivers. Also the load on the servers increases. If multicast is used the media content is sent only once, so the bandwidth is independent of the number of users. This could benefit ISPs and content providers.

The issues regarding multicast deployment or rather the lack of multicast deployment are discussed in [1, 2] and [3]. Basically there are technical reasons and marketing reasons. An interesting point, the so called "three-fold" deadlock [4], is made by the creators of CastGate. Three parties are involved in this situation, ISPs, content providers and customers. The complexity of the protocols involved, the limitations and lack of customer demand have led to the development of several proposals for alternative group communication services (AGCS) [1]. Some make use of tunneling like UMTP [5], overlay multicast like Narada [3] or group specific routing services like Xcast [6].

There are two aspects to native multicast, the host management part and the creation of multicast distribution trees. The first part is done by IGMP [7] for IPv4 and MLD [8] for IPv6, while multicast routing protocols are used to create the distribution trees. PIM (Protocol Independent Multicast) is a multicast routing protocol that is independent of the mechanisms provided by any unicast routing protocol. It requires some unicast routing protocols (such as RIP or OSPF) to determine the network topology and the topology changes. PIM is not a single multicast routing protocol, it has two different modes: PIM-DM (Dense Mode) and PIM-SM (Sparse Mode).

PIM-SM assumes that each receiver has to explicitly join a multicast tree if it wants to receive multicast packets. It creates a core-based tree with a shared root called RP (Rendez-vous Point). The RP is responsible for forwarding all packets destined for the multicast group. Each group has a single RP at any given time, but one RP can serve multiple groups. PIM-SM provides a method for switching to the shortest-path tree, if a certain threshold on a leaf router is exceeded.

2 Alternative Multicast Technologies: CastGate

The CastGate technology is the result of work by the Digital Telecommunications (TELE) research group of the ETRO department at the Vrije Universiteit Brussel. It provides seamless access to multicast content through the use of auto-tunneling [9]. It is intended as a transition technology that will lead to an increase in the number of multicast users, thus forcing ISPs to consider deploying native multicast. It uses a modified version of the UMPT (UDP Multicast Tunneling Protocol) called Enhanced UMTP.

The basic CastGate architecture (Figure 1) consists of three parts: CastGate Tunnel Client (TC), CastGate Tunnel Server (TS) and CastGate Tunnel Database Server (TDS). The database contains information about all the available TSs. Multiple TDSs form what is called a Hierarchical Tunnel Database (HTD). The TS is to be found in the multicast part of the Internet, where it terminates one end of the tunnel. The TC is located at the client side, where it terminates the other end of the tunnel. It will ask the HTD for a list of TSs. The TC informs the chosen

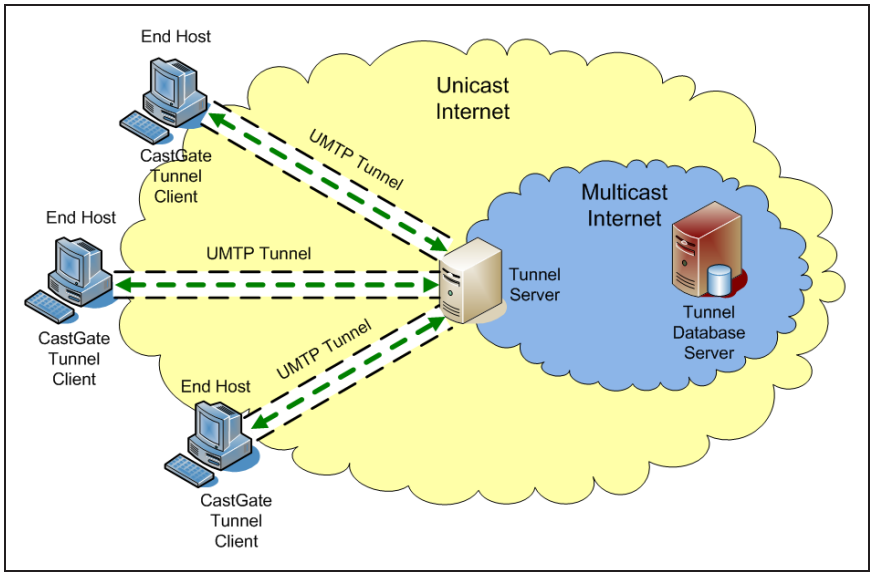


Figure 1: CastGate Client

TS of the multicast group it wants to receive traffic for, and the TS will tunnel the data to the client. The TC can be integrated in a multicast application or it can be a Java applet which runs in a web browser. In either situation, the operation is transparent to the end user. From the client's point of view it is as good as native multicast.

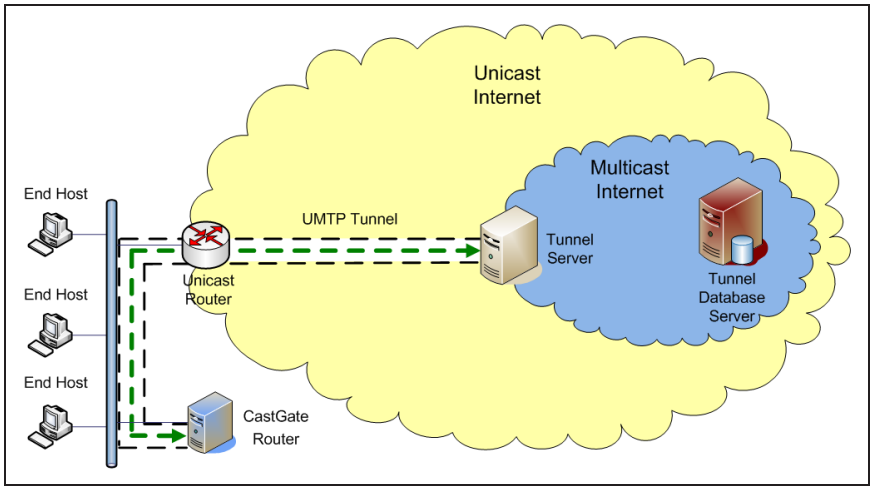


Figure 2: CastGate Router

CastGate Router is a result of the further development of CastGate technology (Figure 2) . It integrates the functionality of an IGMP querier with the Tunnel Client. Thus it provides multicast access to all the hosts on the same LAN segment. The IGMP querier from the CastGate Router keeps track of the group membership for that LAN segment. Based on this information the Tunnel Client will join or leave the multicast group through the tunnel. The advantage of using a CastGate Router is that multicast traffic is tunneled only once for all the receivers on that LAN [10]. The use of the initial technology requires each end host to run a Tunnel Client, thus several unicast packets with identical multicast data are transmitted on the same link.

CastGate allows to address some issues which are not solved by the current IP multicast service. One of them is that native multicast lacks AAA (Authentication Authority Accounting). By adding support for some of the AAA features to Enhanced UMTP, CastGate provides a temporary solution.

To the CastGate project belong CastGuide and CastContent [10]. CastGuide is a session directory tool that allows you to obtain a list of available and upcoming IP Multicast content. It is the equivalent of a TV-guide, but then for Multicast Content. CastContent deals with tools for the content provider, to address certain issues about

access control and accounting.

3 Proposed Improvements: CastGate Router with PIM-SM

Our idea is to extend the functionality of the CastGate Router, so that it can provide multicast access to an entire local domain. Here by domain we understand a group of networks under local administration, where any multicast protocol can be used, but without global multicast access. Tunneling traffic to the local domain with the use of an extended CastGate Router and then distributing that traffic through native multicast would prove a great benefit.

PIM-SM [11] routing protocol is best fitted for the job because it creates multicast delivery trees with a single common root. Information about multicast activity in the domain is gathered by the RP. Placing a modified CastGate Router on the same link as the RP would give us access to information regarding multicast receivers and sources in the domain (Figure 3). The multicast traffic tunneled (by the CastGate Router) to this link will be delivered to all the receivers by the PIM-SM routers without need for further intervention. Also multicast traffic from a source located anywhere in the local domain will reach the RP. Due to implementation complexity it was decided not to embed a PIM-SM router with the CastGate Router, but rather to extract the minimum functionality from the PIM-SM standard [11].

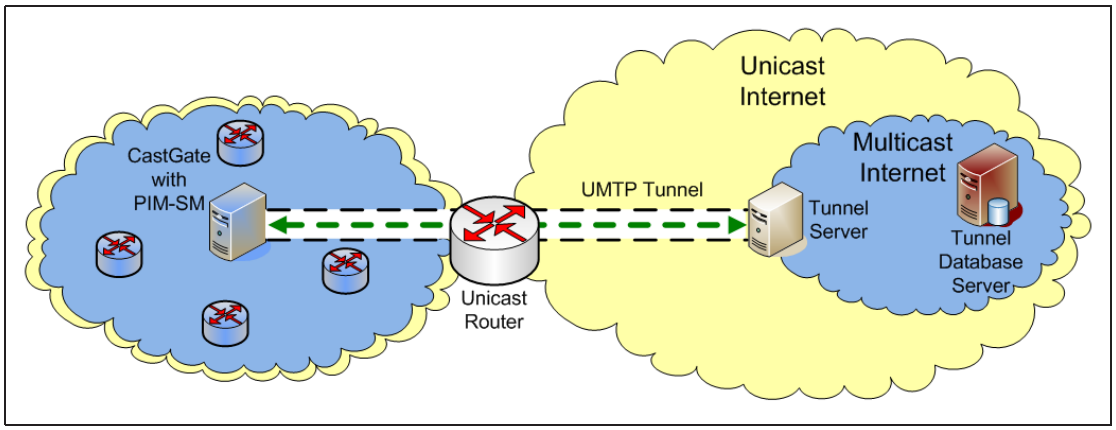


Figure 3: CastGate with PIM-SM

The scenario used is RP-on-a-stick [12]. This happens when the incoming interface of an (S, G) entry at the RP is also the only outgoing interface on the shared tree for group G. It is important to understand that multicast traffic is never forwarded on the same interface it was received on.

3.1 Receiving Multicast from the Internet

The PIM-SM module has to listen to all the messages destined for the RP and it must decide whether to join or leave a group through the tunnel. The module captures PIM-SM messages from which it extracts information about multicast groups that have members in the domain. From the different PIM-SM message types, only two are of interest, Hello and Join/Prune messages. The first type of messages contains information about the neighboring PIM-SM routers on the link, information that will be recorded in a neighbor list. Information about group membership across the domain is contained in the Join/Prune messages (actually (*, G) Join/Prune).

Neighbors will not accept Join/Prune messages from a router unless they have first heard a Hello message from that router. The information from these messages is kept in a list of neighbors on a per interface basis. This list contains the following data: IP address, Holdtime, GenID, LAN Prune Delay (Propagation_Delay (I) and Override_Interval (I)). Holdtime is the amount of time a receiver must keep the neighbor reachable, with a default value of 210 seconds [11]. The GenID option contains a randomly generated 32-bit value that is regenerated each time PIM forwarding is started or restarted on the interface, including when the router itself restarts. When a Hello message with a new GenID is received from a neighbor, any old Hello information about that neighbor should be discarded and superseded by the information from the new Hello message.

Join/Prune messages carry information about the active groups in the domain. The module will listen only to (*, G) Join/Prunes which are used to create core-based trees. These messages specify that group G must be joined or pruned from any source (*). The listener module should check whether the Upstream Neighbor Address and the Joined/Pruned Source Address in the incoming (*, G) Join/Prune message matches the address of the RP (RP address should be configured on the listener module for security reasons).

The PIM-SM module uses a modified version of the downstream per-interface (*, G) state machine from the protocol specifications (Figure 4). Based on the information from the state machine the group will be joined through the tunnel and in this situation multicast traffic from the tunnel is forwarded to the domain.

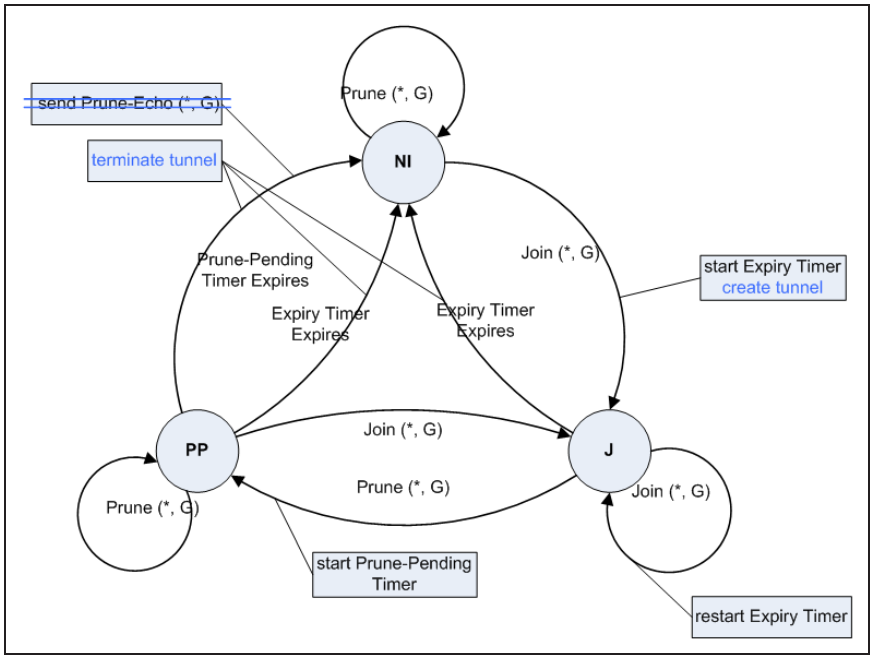


Figure 4: State machine for PIM-SM receiver and PIM-SM

Modifications were necessary because the PIM-SM extension module does not implement the entire functionality of PIM-SM. For example we do not send a (*, G) PruneEcho message when a transition occurs from Prune-Pending to NoInfo state. The differences are marked using another color. Notice the send Prune-Echo (*, G) is crossed out, because it is not used. The transition to Join state determines the creation of the tunnel.

3.2 Sending Multicast from the Local Domain

In order to send multicast traffic from the sources within the local domain, the PIM-SM module must intervene during the Source Registration process. Through this process, the RP is notified of existing multicast sources. When a multicast source begins to transmit, the DR (directly connected to the source) receives the multicast packets sent by the source and encapsulates each one in PIM Register messages. These messages are received by the RP, which de-encapsulates them. The RP will forward the multicast packet down the shared tree and will join the SPT for source S, so that it can receive (S, G) traffic natively. If there is no active shared tree for the group, the RP discards the multicast packets and does not send a Join toward the source. The RP sends PIM Register-Stop messages to the DR, to instruct it to stop sending PIM Register messages.

The module must intercept the PIM Register messages, extract information about the group and de-encapsulate the multicast data which will be tunneled to the Tunnel Server. Join (*, G) messages must be sent to make sure that the RP will join the SPT for the source. Also, in order for the Join message to be accepted by the RP we must send Hello messages [11].

Hello messages are sent periodically. The value for the default Hello Period is 30 seconds. We must take into consideration that these messages are used for the DR election on that link. If the DR_Priority Option is used, the router with the highest value will be the DR. If this option is not present, then the values of the IP address is used to compute the DR [11]. In this case the machine with the highest value is elected. Because our module does not implement the full functionality of a PIM-SM router we must make sure that it is not elected DR on the link. This

can be accomplished by the use of DR_Priority Option with the value set to zero in the Hello messages sent by the module.

The PIM-SM module analyzes captured PIM Register messages. First the Null-Register bit is checked. If this bit is set to 1, then the message is discarded because it contains a dummy header [11]. If the value is 0, then this Register message contains a real multicast data packet. This packet is sent over the tunnel, and also the information regarding the existence of a source for multicast group G is extracted.

The operation is described by a state machine (Figure 5). Once the presence of a source for group G is detected, we must "convince" the RP to join the SPT for source S. This can be accomplished by sending a Join (*, G) message. This message must be sent periodically every 60 seconds. According to standard specification if a PIM router sees a Join (*, G) message on the interface it must suppress its own Join (*, G). Also if it sees a Prune (*, G) it must override it by sending a Join (*, G).

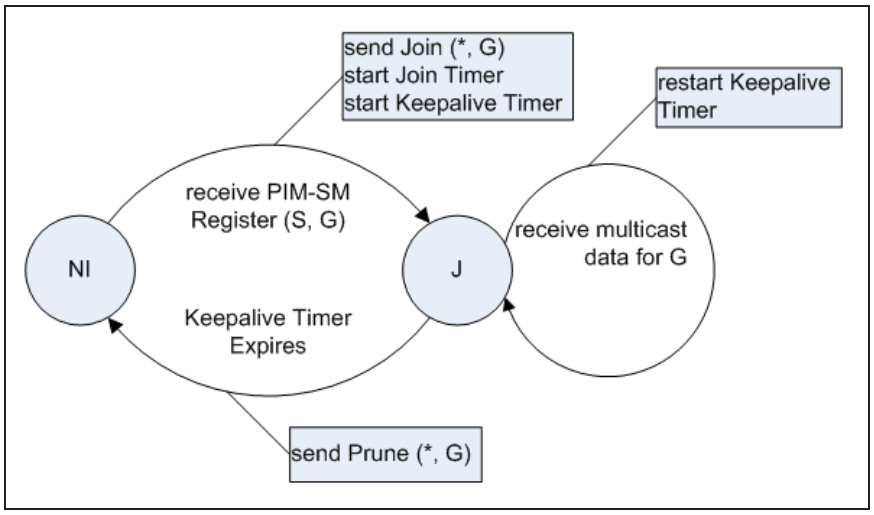


Figure 5: State machine for PIM-SM sender

A keepalive timer on a group basis will be used to decide when to stop sending Join messages. This represents the period after the last data packet for group G was received during which we keep on sending Join messages, and has a value of 210 seconds.

3.3 IPv6 CastGate

The next step toward native IPv6 multicast environment is to have an IPv6 CastGate Router with PIM-SM support. This requires the addition of new functionality to it. The communication between the TC, TS and TDS can be performed using IPv4. Also IPv6 multicast traffic can be tunneled over IPv4. Thus IPv6 unicast connectivity is not required between them. The TDS does not need any modification, but the TC and the TS must be IPv6 capable. CastGate Router would support IPv6 end hosts only if a MLD querier is installed.

To support an IPv6 CastGate version we must implement an IPv6 Enhanced UMTP. Each UMPT datagram contains a 12/16-octet trailer [5]. If we want these trailers to transport IPv6 information, their size has to be modified. This means the 12-octet trailer should be replaced by a 24-octet trailer and the 16-octet one should have 40 bytes.

4 Conclusion

Due to so-called "three-fold" deadlock, the multicast access is not available to the regular Internet user. We proposed an enhancement to the existing alternative solutions (CastGate and CastGate router), i.e. PIM-SM support. This idea could be generalized to other multicast tunneling solutions. Furthermore, we investigated the possibility of using CastGate architectures in IPv6, with a remaining IPv4 tunnel. Obviously once the native IPv6 multicast in fully available, any CastGate-based solutions, no matter its version, will be replaced. This involves a co-operation with the telecom operators for deployment of IPv4/IPv6 multicast services.

Further work will include performance evaluation for the CastGate technology and comparison to native IPv4/IPv6 multicast. The right metrics for AGCS must be determined first. Another issues is how to apply these metrics to native multicast. In the case of multicast, the determination of join latency and control overhead are under progress. Once all the results are available, the penalties of CastGate use will be determined, thus allowing proper technology selection for multicast distribution.

References

[1] Ayman El-Sayed, V. Roca, L. Mathy, "A Survey of Proposals for an Alternative Group Communication Service", *IEEE Network*, January-February, pp. 46-51, 2003.

[2] C. Diot, et al., "Deployment Issues for the IP Multicast Service and Architecture", *IEEE Network*, pp. 78-88, 2000.

[3] Y. Chu, S. Rao, and H. Zhang, "A Case for End System Multicast", *Proceedings of the ACM SIGMETRICS*, 2000.

[4] Pieter Liefooghe, "An Architecture for Seamless Access to Multicast Content", PhD Thesis, *Vrije Universiteit Brussel*, 2002.

[5] Ross Finlayson, "The UDP Multcast Tunneling Protocol", *draft-finlayson-umtp-09.txt*, November 2003.

[6] R. Boivie, et al., "Explicit Multicast (Xcast) Basic Specification", *draft-ooms-xcast-spec-09.txt*, December 2005.

[7] B. Cain, S. Deering, *"Internet Group Management Protocol, Version 3"*, RFC3376, October 2002.

[8] R. Vida, L. Costa, *"Multicast Listener Discovery Version 2 (MLDv2) for IPv6"*, RFC3810, June 2004.

[9] Pieter Liefooghe, "CastGate: An Auto-Tunneling Architecture for IP Multicast", *draft-liefooghe-castgate-02.txt*, October 2004.

[10] Pieter Liefooghe, M. Goossens, A. Swinnen, B. Haagdorens, "The VUB Internet Multicast "CastGate" Project", Technical Report 10/2004 v1.8, *Vrije Universiteit Brussel*.

[11] Bill Fenner, M Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", *draft-ietf-pim-sm-v2-new-11.txt*, 25 October 2004.

[12] Beau Williamson, *Developing IP Multicast Networks, Volume 1*, Cisco Press, 2001.

Tudor Mihai Blaga, Virgil Dobrota, Gabriel Lazar, Bogdan Moraru
Technical University of Cluj-Napoca
Communications Department
Address: 26-28 Baritiu St., 400027 Cluj-Napoca, Romania
E-mail: {tudor.blaga, virgil.dobrota,gabriel.lazar,bogdan.moraru}@com.utcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 86-91

String Comparison in Terms of Statistical Evaluation Applied on Biological Sequences

Alina Bogan-Marta, Nicolae Robu, Mirela Pater

Abstract: Protein sequences from all different organisms can be treated as texts written in a universal language where the alphabet consists of 20 distinct symbols, the amino-acids. The mapping of a protein sequence to its structure, functional dynamics and biological role then becomes analogous to the mapping of words to their semantic meaning in natural languages. This analogy can be exploited by applying statistical language modeling and text classification techniques for the advancement of biological sequences understanding. Here a new general strategy for measuring similarity between proteins is introduced. Our approach has its roots in computational linguistics and the related techniques for quantifying and comparing content in strings of characters. We experimented with different implementations having as ultimate goal the development of practical, computational efficient algorithms. The experimental analysis provides evidence for the usefulness and the potential of the new approach and motivates the further development of linguistics-related tools as a means to decipher the biological sequences.

Keywords: n-grams, entropy, cross-entropy, protein similarity, exploratory data analysis.

1 Introduction

The practice of comparing gene or protein sequences with each other, in the hope of elucidating similarity conveying functional and evolutionary significance, is a subject of primary research interest in bioinformatics.The rewards range from the purely technical, such as the identification of contaminated sequence phases, to the most fundamental ones, such as finding how many different domains define the three of life.

The most frequently used methods for measuring protein similarities are based on tedious algorithmic procedures for sequence alignment. I our days there are a large variety of methods like: Smith-Waterman dynamic programming algorithm [1](considered as standard reference method due to the accuracy of the obtained results), heuristic algorithms[1], methods based on hidden Markov models [2],[3].Some of the main concepts identified in new alternative methods proposed for sequence similarity are Dirichlet mixtures [4], sliding windows technique [5], a mixture model of common ancestors [6], support vector machines (SVMs)[7], latent semantic analysis (LSA)[8].

Despite the maturity of the developed methodologies working towards this direction, the derivation of protein similarity measures is still an active research area. The interest is actually renewed, due to the continuous growth in size of the widely available proteomic databases that calls for alternative cost-efficient algorithmic procedures which can reliably quantify protein similarity without resorting to any kind of alignment. Apart from efficiency, a second specification of equal importance for the establishment of similarity measures is the avoidance of parameters that need to be set by the user (a characteristic inherent in the majority of the above mentioned methodologies). It is often the case with the classical similarity approaches that the user faces a lot of difficulties in the choice of a suitable search algorithm, scoring matrix or function as well as set of optional parameters whose optimum values correspond to the most reliable similarity.

Here, a new approach for measuring the similarity between two protein sequences is presented. It was inspired by the successful use of the entropy concept for information retrieval in the field of statistical language modeling [9],[10]. Although the *n*-gram concept has been used in earlier works, e.g.[11],[12] the presented one is a new attempt to adopt this dual step for comparing biological sequences.

For the thorough validation of the suggested similarity measure, we used a corpus of sequences built from a publicly available database. Using standard procedures, well-known in the field of exploratory data analysis and information retrieval, we evaluated the performance of our measure. We show that this new method provides an effective way for capturing the common characteristics of the compared sequences, while avoiding the annoying

[1]Basic Local Alignment Search Tool, http://www.ncbi.nlm.nih.gov/BLAST/
FAST-All, or fast protein/nucleotide comparison, http://www.ebi.ac.uk/fasta33/
CLUSTAL, http://www.ebi.ac.uk/clustalw/'.

task of choosing parameters, additional functions or evaluation methods. This high performance and the ready-to-plug-in character, taken together with its computational efficiency, make our approach a promising alternative to the well-known, sophisticated protein similarity measurements.

2 Background Concepts

There are various kinds of language models that can be used to capture different aspects of regularities of natural language [13]. Markov chains are generally considered among the more fundamental concepts for building language models. In this approach the dependency of the conditional probability of observing a word w_k at a position k in a given text is assumed to be depended only upon its immediate n predecessor words $w_{k-n} \dots w_{k-1}$. The resulting stochastic models, usually referred as n-grams, constitute heuristic approaches for building language grammars and their linguistic justification has often been questioned in the past. However, in practice they have turned out to be extremely powerful. Nowadays n-gram modeling stands out as superior to any formal linguistic approach [14] and has gained high popularity due to its simplicity. Closely related with the design of models for textual data are algorithmic procedures for validating them. Entropy is a key concept for this kind of procedures. In general, its estimation is considered to provide a quantification of the information in a text and has strong connections to probabilistic language modeling [15]. While relying on the same theoretical principles, the estimation of entropic-measures in the domain of language processing requires some modifications (dictated by the discrete nature of data) with respect to the procedures established in the field of statistics. As described in [10] and [16], the entropy of a random variable X that ranges over a domain $\aleph$, and has a probability density function, P(X) is defined as

$$H(X) = -\sum_{X \in \aleph} P(X) log(X). \tag{1}$$

The cross-entropy between the actual probability distribution P(X) (over a random variable X) and the probability distribution Q(X) estimated from a model is defined as follows

$$H(X,Q) = -\sum_{X \in \aleph} P(X) logQ(X). \tag{2}$$

Two important (for the development of our approach) propositions are mentioned here. First, the cross-entropy of a stochastic process, measured by using a model, is an upper bound on the entropy of the process (i.e. H(X)=H(X,Q)) [9],[16]. Second, as mentioned in (19), between two given models, the more accurate is the one with the lower cross-entropy. In [17], the general idea of entropy has been adopted in the specific case that a written sequence $W = w_1, w_2, \dots, w_k - 1, w_k, w_{k+1}, \dots$ is treated as an n-gram based composition and resulted in the following estimating formula

$$H(X) = -\sum_{W^*} p(w_1^n) log_2 p(w_n | w_1^{n-1}) = -\frac{1}{N} \sum_{W^*} Count(w_1^n) log_2 p(w_n | w_1^{n-1}), \tag{3}$$

where the variable X has the form of an n-gram $w_1^n = w_1, w_2, \dots, w_n$, the summation runs over all the possible n-length combinations of consecutive w (i.e. $W^* = \{\{w_1, w_2, , w_n\}, \{w_2, w_3, \dots, w_{n+1}\}, \dots\}$ and $\boldsymbol{N}$ is the total number of n-grams in the investigated sequence. The second term in the summation is log_2 from the conditional probability that relates the n^{-th} element of an n-gram with the preceding n-1 elements. Following the principles of maximum likelihood estimation (MLE), it can be a counting procedure expressing the corresponding relative frequencies:

$$P(w_n | w_1^{n-1}) = \frac{Count(w_1^n)}{Count(w_1^{n-1})} \tag{4}$$

The above entropic estimation (taken together with the general form of equation 1 and 2 suggesting a direct way to pass from entropy to cross-entropy formulation) was the basis for building our similarity measure, described in the sequel.

3 Linguistic Approach of the New Protein Similarity Measure

Due to the text representation of biological sequences, the mapping of a protein sequence to its structure, functional dynamics and biological role then becomes analogous to the mapping of words to their semantic meaning

in natural languages. Scientists within this hybrid research area have become optimist about the identification of Gramar/Syntax rules that could reveal systematics of high importance for biological and medical sciences. In the presented method, we adopted a Markov-chain grammar and built for our protein dataset 2-gram, 3-gram and 4-gram models for each protein sequence. To clarify things let a protein sequence WASQVSENR. In the 2-gram modeling the available "words" are WA AS SQ QV VS SE EN NR, while in the 3-gram representation the words are WAS ASQ SQV QVS VSE SEN ENR. Based on the frequencies of these words (estimated by counting) and by forming the appropriate ratios of frequencies, the entropy of a *n*-gram model can be readily estimated with (3). This measure is indicative about how well-predicted is a specific protein sequence by the corresponding model. While this measure could be applied for two distinct proteins (and help us to decide about which protein is better represented by the given model), the outcomes couldn't facilitate the direct comparison of the two proteins (and help us to decide if they are similar or not). The previous shortcoming led us to devise the corresponding cross-entropy measure, in which the *n*-gram model is, first, built based on the word-counts of one protein sequence (training-step) and then the predictability, of the second sequence, by the model is measured (projection-step) as a means of contrasting the two proteins. So the common information content is expressed via the formula

$$E(X,Y) = -\sum_{w_i^n \in X} P_X(w_i^n) log P_Y(w_{i+n}|w_i^{n-1}) \tag{5}$$

The first term, $P_X(w_i^n)$ in the above summation, refers to the reference protein sequence X (i.e. it results from counting the words of that specific protein). The second term corresponds to the sequence Y based on which the model has to be estimated (i.e. it results from counting the words of that protein).Variable w_1^n ranges over all the words of the reference protein sequence.

3.1 Database Searches with the New Similarity Measure

Having introduced the new similarity measure, we proceed here with the description of its use for performing searches within protein databases. The main aspect of our approach is that both the unknown query-protein (e.g. a newly discovered protein) and each protein in a given database (containing annotated proteins with known functionality, structure etc.) are represented via *n*-gram encoding and the above introduced similarity is utilized to compare their representations. By recognizing the most similar proteins within the database, the structure and function of the query-protein can be inferred based on the principle of assimilation. In the algorithmic implementation of these ideas, and as a byproduct of the experimentation with actual data, we devised two different ways in which the *n*-gram based similarity is engaged in efficient database searches. The most direct implementation lead us to an algorithm called hereafter as ***direct method***. A second algorithm, the ***alternating method***, was devised in order to cope with the fact that the proteins to be compared might be of very different length.

Direct method. Let S_q the sequence of a query-protein and $S = S_1, S_2, \ldots, S_N$ the given protein database. The first step is the computation of 'perfect' score (*PS*) or 'reference' score for the query-protein. This is done by computing $E(S_q, S_q)$ using the query-protein both as reference and model sequence (we call here "model" the sequence compared with the query) in equation 5. In the second step, each protein S_i, i=1,...,N, from the database serves as the model sequence in the computation of a similarity score $E(S_q, S_i)$, with the query-protein serving as reference sequence. In this way, N similarities are computed $E(S_q, S_i)$, i=1,..,N. Finally, these similarities are compared against the perfect score *PS* by computing the absolute differences $D(S_q, S_i) = |E(S_q, S_i) - PS|$. The 'discrepancies' in terms of information content between the query-protein and the database-proteins are expressed. By ranking these N measurements, we can easily identify the most similar proteins to the query-protein as those which have been assigned the lowest distance $D(S_q, S_i)$.

Alternating method. The only difference with respect to the direct method is that when comparing the query-protein with those from the database, the role of reference and model protein can be interchanged based on the shortest (the shortest sequence plays the role of reference sequence in equation 5). The other steps, perfect-score estimation, ranking and selection, follow as previously.

4 Experiments

The proposed strategy for measuring protein similarity was demonstrated and validated using a database containing an overall sample of 100 protein sequences. Two distinct groups of protein data were selected as follows.

The first 50 entries of the database corresponded to proteins selected at random from the NCBI public database [2]. The last 50 entries corresponded to proteins resulted from different mutations of the p53 gene. The mutations were selected randomly from the database we created using the descriptions, provided by the International Agency for Research on Cancer (IARC) Lyon, France [3]. This set of 50 proteins, denoted hereafter as p53-group, is expected to form a tight-cluster of textual-patterns in the space of biological semantics. On the contrary, the rest of 50 proteins should appear as textual-patterns in the same space that differ not only with the other, but also (and mainly) from the p53-group.

4.1 Results

In order to provide quantitative measures of performance for the two variants, we adopted an index of search accuracy, that is derived from receiver operating characteristic (ROC) curves and has recently gained popularity when validating protein-databases searches [18]. This index, usually referred as truncated ROC-score, is the ratio of the area under the ROC-curve (in the plot of true- positives versus false positives for different thresholds of dissimilarity). More explicitly, for a number T of true positives available to be found and a fixed number of false positives *n*, this index is the proportion of the rectangle [0, T] [0, n] that lies under the sensitivity curve. It takes values in the range [0-1], with one corresponding to the highest performance. This ROC-score has been tabulated in Table 1 for different *n*-grams and both methods.

n-gram	Thresholds	Direct method		Alternating Method	
		FPR	TPR	FPR	TPR
2-gram	0.005	0.009	0.298	0.009	0.416
	0.1	0.106	0.698	0.118	0.849
	1.46	0.900	1.000		
	1.74			0.891	1.000
3-gram	0.005	0.007	0.476	0.018	0.512
	0.1	0.256	0.701	0.250	0.834
	0.38	0.885	1.000	0.895	1.000
4-gram	0.005	0.005	0.527	0.005	0.612
	0.1	0.005	0.894	0.007	0.999
	0.25	0.175	1.000		

Table 1: False positive rates and true positive rates for different *n*-grams using both evaluation strategies.

In addition, following some classical steps of Exploratory Data Analysis we obtained the matrix containing all possible dissimilarity measures $D(S_i, S_j), i, j = 1, 2, \ldots, N$ for the model of 4-gram using the *alternating method*. The results are plotted in Figure 1 which is a low-dimensional representation of protein sequences based on the dissimilarity values. As can be easily observed, the sequences belonging to the two defined groups of proteins (unknown and mutated) are clearly separated. More than that, we identified the small subgroups of mutated proteins as representing the similar sequences with close length.

5 Conclusions

Within this paper we specifically studied the use of cross-entropy derived measure applied over *n*-gram models as a means of searching in protein database in an effective and efficient way. The experimental results indicated the reliability of our algorithmic strategy for expressing similarity between proteins. Given the conceptual simplicity of the introduced approach, it appears as an appealing alternative to previous well-established techniques.

During the evaluation of our method we observed that from the two introduced variants the better performance is associated with the second one. This means that it is important to come up with improvements that overcome the possible wrong identifications of similar sequences due to the decisively big differences between sequences length. In the exceptional case when all the compared sequences have the same length, the direct method is equivalent with

[2] National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov/)

[3] http://www.iarc.fr/

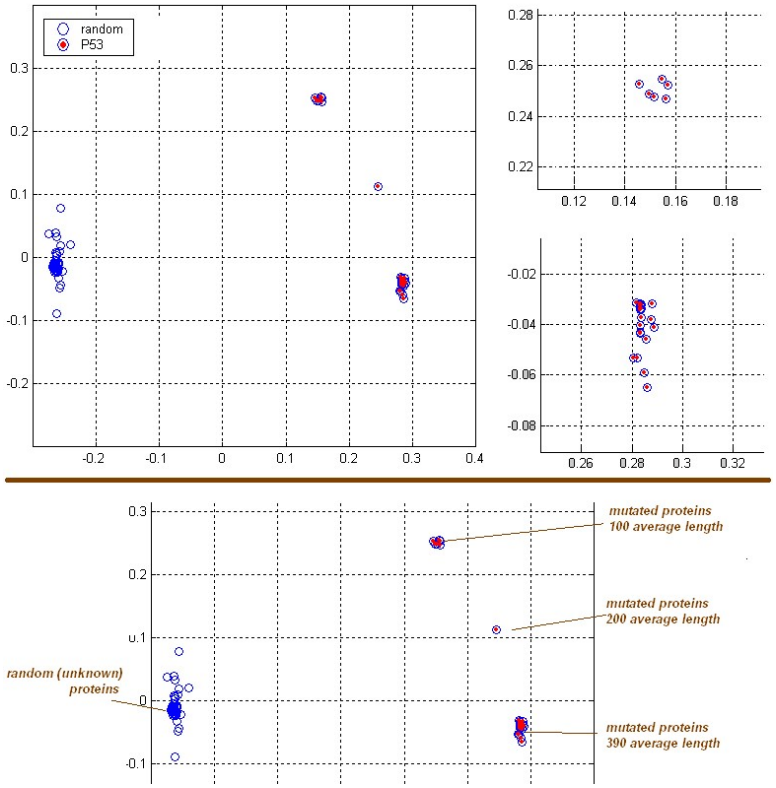


Figure 1: Low-dimensional representation of protein sequences using dissimilarity measure for experimental data

the alternating method and performs excellent. Regarding the order of the employed n-gram model, after testing with order of 2,3,4,5 we noticed that the performance of the method increases with the order of the model up to 4. After the order of 5 due to lack of data the corresponding maximum likelihood estimates becomes unreasonable uniform and very low. This sets an upper limit for our model order in the specific database (perhaps slightly higher order model could work in different protein databases.

The groups identified in Figure 1 are giving a new perspective over the clustering potential of the method as long as it is easy to observe the distinct groups of sequences differentiated by their content and length. For particular application of detection of mutated sequences, it can be considered the problem of two class identification: mutation and unknown(random) sequences. More than that, the subgrups of mutated sequences deserve more attention in order to identify possible functional related sequences.

Before continuing the work on the improvement of this method we have to remark that this is a statistical in nature technique. It can be improved by incorporating biological knowledge (e.g. working with functional groups of amino-acids). Finally, another aspect that deserves further consideration is to test if our method scales well with the size of the protein database.

6 Acknowledgement

This work was supported by the EU project Biopattern: Computational Intelligence for biopattern analysis in Support of eHealthcare, Network of Excellence Project No. 508803.

References

[1] T. Smith, M. Watermann ,“Identification of common molecular subsequences,” *J. Mol. Biol.*, Vol. 147, pp. 195-197, 1981.

[2] A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler,"Hidden Markov models in computational biology: Application to protein modeling",*J. Mol. Biol.*, Vol.235, pp.1501-1531, 1994.

[3] P. Baldi, Y. Chauvin, T. Hunkapiller, and M.A. McClure, "Hidden Markov models of biological primary sequence information",*in Proc. Natl. Acad. Sci. USA*, Vol.91(3), pp.1059-1036, 1994.

[4] K. Sjolander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I.S. Mian, and D. Haussler, "Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology,"*J. Bioinformatics*, Vol 12, pp: 327-345, 1996.

[5] M.V. Katti, R. Sami-Subbu, P.K. Ranjekar and V.S. Gupta,"Amino acid repeat patterns in protein sequences: Their diversity and structural-functional implications,"*Protein Sci.*, Vol. 9, pp: 1203-1209, 2000.

[6] E. Eskin, W.N. Grundy and Y. Singer, "Using mixtures of common ancestors for estimating the probabilities of discrete events in biological sequences,"*Bioinformatics*, Vol.17 Suppl 1, pp: 65-73, 2001.

[7] H. Saigo, J-P. Vert, N. Ueda, and T. Akutsu, "Protein homology detection using string alignment kernels," *J. Bioinformatics*, Vol.20 no.11, pp:1682-1689, 2004.

[8] M.K. Ganapathiraju, J. Klein-Seetharaman, N. Balakrishnan and R. Reddy,"Characterization of protein secondary structure-application of latent semantic analysis using different vocabulary,"*IEEE Signal Processing Magazine*, Vol. 21, no.3, pp: 78-87, 2004.

[9] C.D. Manning, and H. Schütze, 2000, *Foundations of statistical natural language processing*, Massachusetts Institute of Technology Press, Cambridge, Massachusetts London, England,.554 - 556;557 - 588.

[10] D. Jurafsky, and J. Martin,*Speech and Language Processing*, Prentice Hall, 2000, pp: 223-231.

[11] M. Ganapathiraju, V. Manoharan, and J. Klein-Seetharaman,"Statistical sequence analysis using n-grams",*J. Appl. Bioinformatics*, Vol.3 (2), pp:193-200 , 2004.

[12] S. Karlin, and C. Bruge,"Trinucleotide repeats and long homopeptides in genes and proteins associated with nervous system disease and development",*Proc Natl Acad Sci USA*, Vol. 93(4), pp:1560-1565, 1996.

[13] S. Wang, D. Schuurmans, F. Pengun, and Y. Zhao," Semantic N-gram Language Modeling With The Latent Maximum Entropy Principle",*In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03)*, http://citeseer.nj.nec.com/575237.html.

[14] D. Van Compernolle, "Spoken Language Science and Technology", 2003.

[15] R. Durbinn, S. Eddy, A. Crogh, G. Mitchison, *Biological sequence analysis.*in Probabilistic models of proteins and nucleic acids, Cambridge University Press, 1998.

[16] P.F. Brown, A. S. Della Pietra, V.J. Della Pietra, L.R. Mercer Robert, and C.L. Jennifer, "An estimation of an upper bound for the entropy of English", *in Association for Computational Linguistics*, Yorktown Heights, NY 10598, P.O. Box 704, 1992.

[17] D.H. Van Uytsel, and D.Van Compernolle, D.,"Entropy-based context selection in variable-length n-gram language models", *IEEE Benelux Signal Proc. Symp.*, pp. 227-230, 1998.

[18] A. Schäffer, L. Aravind,L. Madden,S. Shavirin,J. Spouge, Y. Wolf, E. Koonin, S. Altschul,*Nucleic Acids Research*, Vol. 29(14), pp: 2994-3005, 2001.

Alina Bogan-Marta, Mirela Pater
University of Oradea, Department of Computer Science
Address: 1, Universitatii St., Oradea
E-mail: {alinab,mirelap}@uoradea.ro

Nicolae Robu
"Politehnica" University of Timisoara, Department of Automatics
Address: 2, Vasile Parvan
E-mail: nrobu@aut.utt.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 92-100

Formal Modeling of Concurrent AOP Programs

Crenguţa Mădălina Bogdan, Luca Dan Şerbănaţi

Abstract: In the last few years, a new trend in programming centered on a new paradigm called aspect-oriented programming(AOP) has emerged. AOP is based on the object-oriented programming and introduces a new concept, the aspect. The aspects model those features being in many objects of an application may change or evolve independently one from the others.
Because of this independence, it is not easy to verify the correctness of the AOP programs. Moreover, the problem of verifying the correctness is harder and more important in the case of the concurrent AOP programs than ordinary ones. The paper claims that formal specification of concurrent AOP programs could be a powerful tool for their correctness verification. Such specifications have the advantage that can be used to demonstrate important properties of AOP programs, like safety and liveness properties.
The truth of some of these properties is formally proved on the AOP solution of the Producer-Consumer problem. Its specifications are constructed using the Temporal Logic of Actions. The program is written in AspectJ, the most popular language implementing the AOP paradigm.

Keywords: aspect-oriented programming, concurrency, correctness, formal specification

1 Introduction

Aspect-oriented programming (AOP) is a new programming paradigm that emerged with few years ago from the need to deal with nonfunctional crosscutting properties of programs. Crosscutting features including access control, synchronization policies and resource sharing over objects are modularized as aspects that are separate entities from objects.
In AOP we have two kind of units of modularization: objects, which represent the program functionality, and aspects that resolve those properties or features which affect the performance or other quality attributes of the program.
An aspect weaver, which is a compiler-like entity, weaves objects and aspects together into a program, using a mechanism of join points. A joint point is a well-defined point into the program flow. Execution points of method invocation and exception throwing may be specified as join points. In such a join point, a piece of code from an aspect should be executed when the control flow will reach at that point.
There are several AOP languages such as AspectC++ and AspectJ (see [2]), which are aspect-oriented extensions to C++ and Java, respectively. In this paper, we use AspectJ.
In AspectJ, a join point is picked out by *pointcut* and *advice* brings together a pointcut and a body of code that will be executed at each join point of the pointcut. If a joint point indicates an object method call, then advice will append the code before/after/around the method call and will modify the method execution.
Like Java, AspectJ allows implementation of the generalization/specialization relationship with some restrictions, though: aspects may only extend abstract aspects, extending a class does not provide the ability to instantiate the aspect with a not null expression, and a class can not extend nor implement an aspect.

1.1 Related Work

In general, the problem of demonstration of correctness of the concurrent program written in a programming language was a concern in the programmers world. This statement is confirmed by the hardworking research undertaken in this domain and the specialty rich bibliography. We cite here only few important papers ([6, 8, 9]).
In the case of AOP programming, this problem becomes more acute because of fact that AOP programs use aspects that contain program sequences that are executed in some (join) points of execution flows of programs.
Although the domain is new, some research was done until now about the formal approach of AOP programs. There are papers that focus on formal defining of semantics of concepts of AOP or sub languages of AspectJ. For instance, Douence et al. have proposed a formal definition of crosscuts which uses execution monitors as an

operational model for crosscutting [1]. Jagadeesan et al. have proposed an operational semantics for the core of AspectJ, incorporating several different kinds of pointcuts and advice in an object-oriented setting [4]. Walker et al. proposed a much simpler formal model incorporating just the lambda calculus, advice, and labeled hooks that describe where advice may apply [10]. As a foundational calculus, their model is ideal for studying compilation strategies for AOP languages.
Regarding the formal correctness of concurrent AOP programs, after our knowledge this research domain was not yet approached. We hope that this paper will constitute a good start in this research field.
The paper is structured as follows. In the next section we give a program that resolves the Producer-Consumer problem using the concepts of AOP programming and the AspectJ language. Section 3 presents the TLA formal specifications of the AOP program. Using the TLA specifications we prove in the section 4 that our program verifies two properties: the mutual exclusion of the producer and consumer and, in some conditions, a given action is eventually executed by the program. We finish the paper with a conclusions section.

2 An AOP Solution of Producer-Consumer Problem

We will present an AOP solution of the Producer-Consumer problem, in the case of one producer and one consumer. As we know, a producer puts data into a buffer and a consumer gets the data from the buffer.
For sake of simplicity, we assume that the data produced and consumed are characters. The producer puts data in only if the buffer is not full. Otherwise, the producer must wait until the buffer is not full. The consumer gets data only if the buffer is not empty. In the Figure 1 we give a solution of this problem using AspectJ. We can observe that in the class Buffer, the only methods which modifies the instance variables of the class are *put* and *get*. The other two methods, getLast and getLength, implement queries on the values of the *last* variable and on the length of the *b* array, respectively, and do not change them.
As it is well known, the using of concurrency raises two problems: mutual exclusion and conditioned synchronization of the processes (threads) that execute or use some shared sequence of instructions or data. The first problem was resolved by the Java language and was taken over by AspectJ through synchronizing the access to the methods that use the shared data. In the case of the Buffer class, the methods *put* and *get* are declared as *synchronized* in their prototype.
The second problem is resolved by three aspects. The first aspect is abstract, because it declares two semaphore variables *prodWait* and *consWait* that will not permit in some cases the access of the producer and consumer, respectively, to buffer.
The tries of the producer and consumer to access the buffer will be caught through the *bufferPut* and *bufferGet* pointcuts by two aspects: *CondSinPut* and *CondSinGet*. The former aspect deals with the extreme case in which the buffer is full and the producer (that is, the current thread) wants to put a character in buffer. In order to do that, the producer must wait before it call the method *put*. In this way, the *CondSinPut* aspect resolves the problem of conditioned synchronization from the viewpoint of the producer. In a similar way, the problem is resolved by the *CondSinGet* aspect for the consumer that has to wait before to call the method *get*, if the buffer is empty. The consumer will be "announced" by the *CondSinPut* aspect after the producer has put a character in buffer.

3 Formal specification of AOP Programs

Since the aspects modify the behavior of objects in a AOP program, we may specify such a program in its behavioral terms. A *behavior* is a sequence of those states through the program has passed while its instructions have been executed. If an AOP program is basically concurrent, that is it contains objects that are threads, the program may have several behaviors generated by the non-deterministic interleaving of (finite or infinite) atomic operations belonging to the thread algorithms.
We will say that an AOP program *verifies* a property if the property is fulfilled by the behaviors representing all correct program executions. We can associate a property of a program with the set of the program behaviors verifying it. As a program specification is a property, we can identify it with the set of behaviors representing all correct executions of the program. Then a program specification is a property, namely the set of behaviors representing a correct program execution.
As it is known, a kind of properties has emerged from proving correctness of concurrent programs. These properties have been classified in two classes: safety and liveness properties, respectively. A *safety property* asserts that a program never enters in a unacceptable state during its execution. Mutual exclusion, partial correctness are

examples of safety properties. A *liveness property* asserts that a program eventually enters in a given state during its execution. By example, the program termination or some action is eventually executed or eventually impossible to execute are liveness properties. In the literature, it is shown that many liveness properties of a program are true only if the planning mechanism of instructions execution is fair. As a definition of fairness, we use a general one given in [9], namely: a concurrent program is fair if and only if each process of the program has a chance to progress, does not matter what the other processes do.
Formally, a property is a predicate on behaviors of a program. Furthermore, the properties can be expressed as formulas in a temporal logic.
There are many temporal logics in the literature (see [3]), like Propositional Linear Temporal Logic, First-Order Linear Temporal Logic, Computational Tree Logic and Temporal Logic of Actions (TLA). In this paper, we will use TLA.

3.1 Temporal Logic of Actions

TLA ([7]) was introduced by L. Lamport in 1990 as a simple variant of Pnueli's original logic, in order to construct specifications of the concurrent programs. The Lamport's goal was to specify and verify the discrete systems in terms of their actions. That is why, TLA can be decomposed in two logics: the logic of actions and the basic temporal logic. The deductive system of TLA belongs to the later logic.
The deductive system is formed from axioms and inference rules. We make no attempt to give the complete set of rules; we just present the ones that we will use in proofs of the properties from the section 4.
Invariance proofs of safety properties are based on the following two rules:

$$\text{Imply Generalization Rule} \quad \frac{T_1 \Rightarrow T_2}{\Box T_1 \Rightarrow \Box T_2}$$

$$\text{Invariance Rule} \quad \frac{I \wedge [Next]_{var} \Rightarrow I'}{I \wedge \Box[Next]_{var} \Rightarrow \Box I}$$

where formula $\Box T_1$ says that the temporal formula T_1 is always true during a behavior of the analyzed program. $[Next]_{var}$ is defined to mean $Next \vee var' = var$, that is either $Next$ holds or the variables of program do not change their values during the execution step. $Next$ is defined as the join of the predicates given by the program actions definitions.
The demonstration of liveness properties may be always reduced to the proof of the properties "leads-to", i.e. formulas by form $P \longmapsto Q$, where P and Q are TLA predicates. Leads-to properties are derived from weak fairness assumptions by using the following rule, where P and Q are predicates and $Next$ and A are actions:

$$\text{Weak Fairness Rule} \quad \frac{\begin{array}{c} P \wedge A \Rightarrow Q' \\ P \wedge Next \wedge \neg A \Rightarrow (P' \vee Q') \\ P \Rightarrow Enabled(A) \end{array}}{\Box[Next] \wedge WF(A) \Rightarrow (P \longmapsto Q)}$$

where $Enabled(A)$ is a predicate which is true in a state s if and only if the action A can take place in s. TLA uses the notation $WF(A) \stackrel{not}{=} \Box\Diamond A_{var} \vee \Box\Diamond\neg Enabled(A)$. This formula expresses the condition of weak fairness, that is the action A is eventually executed or eventually is impossible to execute it. Similarly, we use the notation $SF(A) \stackrel{not}{=} \Box\Diamond A_{var} \vee \Diamond\Box\neg Enabled(A)$ for strong fairness condition: the action A is eventually executed or its execution is infinite often impossible. $P \longmapsto Q$ is equal by definition with $\Box(P \Rightarrow \Diamond Q)$. This formula asserts that whenever P is true, Q is true in that moment or in a subsequent moment of time, that is $\Diamond Q$ is true.

3.2 Specifying AOP programs in TLA

As we stated in a previous subsection, there is no difference between a program specification and a TLA temporal formula. Lamport proposed in [7] the following pattern to construct a temporal formula using TLA:
1. Choose the variables, the type invariant and initial predicate $Init$. This predicate contains assertions about the initial values of program variables. Also, we may define some additional constants.
2. Write the next-state action $Next$. In order to do that, we have to decompose the program into atomic actions and define them. Then $Next$ will be given by the join of the predicates given by the program actions definitions.
3. Write the liveness property V of the program, which represents the temporal part of the specification. V is equal

with conjunction of the fairness conditions (usually, weak) of the next-state action and thus specifies the program's progress condition.
4. Combine the previous defined elements into a single temporal formula that is the specification.
By applying the pattern just described we obtain the following temporal logic formula:

$$\exists v_1 \ldots v_n : \; Init \wedge \Box[Next]_{var} \wedge V$$

where *var* contains all the variables of program.
Note that the formula $\Box[Next]_{var}$ asserts a safety property of the program: each execution step is the execution of a legitimate step defined by the specification. In addition, the progress condition describes what the program eventually must do.
Furthermore, we will apply the previous described pattern to construct a specification of the AOP program for the Producer-Consumer problem. Basically, the program consists from the class Buffer and three aspects. In the followings we will construct specifications for the class and each aspect.
For the class Buffer, we define a type invariant to assert that the variables $last$, pc_{prod} and pc_{cons} are of Nat type. These variable represent the position in buffer on which will be placed the current element and the control state of the producer and consumer, respectively. The predicate $Init$ asserts that initially the buffer is empty and the variables $last$, pc_{prod} and pc_{cons} have the value 0. The enable condition of the action $put(c)$ is: the value c is of $Data$ type and the buffer is not full. In a similar way, the enable condition of the action pop asserts that the buffer is not empty. The predicate $Next$ specifies that each modification of the buffer state is caused by a call of put or get method. The temporal formula $Spec_{buffer}$ specifies assertions about the properties of the buffer: begins empty, every and each modifying of its state is caused by a put or get action (that is, the predicate $\Box[Next]_b$ is true), and always if there is something in the buffer, eventually a *get* must happen (the weak fairness property $WF_b(get)$). We note that we don't request that any element should ever put in buffer. Similarly, we can construct TLA specifications for the aspects of our AOP program. The specifications are presented in the figures 2-4.
Finally, the program itself is the temporal logic formula $Spec$ defined by
$Spec == Spec_{buffer} \wedge Spec_{condSinPut} \wedge Spec_{condSinGet}$, where $Next_1 = put(c) \vee b_2$, $Next_2 = get \vee c_2$, $Next = Next_1 \vee Next_2$ and $V = WF(get) \wedge SF(b_2) \wedge SF(c_2)$.

4 Proving Correctness of AOP Programs

Any property of a program can be expressed by a TLA formula. In addition, the assertion that "The Spec specification has the property P" is expressed in TLA through the validity of the formula $Spec \Rightarrow P$, where P is a TLA predicate.

4.1 Proving Safety Properties

As we stated in Section 2, any AOP concurrent program must fulfill the property of mutual exclusion, that is at most one process at a time can have control at 7, 10, 11, or 16. Formally, the property is enunciated and proved in the following proposition.

Proposition 1. *The AOP program of the Producer-Consumer problem has the property* $\Box\neg\left(pc_{prod} \in \{7,10\} \wedge pc_{cons} \in \{11,16\}\right)$.

Proof. We note with P the predicate $\neg\left(pc_{prod} \in \{7,10\} \wedge pc_{cons} \in \{11,16\}\right)$. In order to prove the formula $Spec \Rightarrow \Box P$, we have to find an invariant I that satisfies three properties: (i) Init⇒I; (ii) I⇒P; and (iii) I∧$\Box[Next]_{var}$ ⇒I', where $Next = put(c) \vee get \vee b_2 \vee c_2$, that is $Next$ is formed from the atomic operations of our program, and $var =< b, last, prodWait, consWait, pc_{prod}, pc_{cons} >$. The following proof shows that these three properties imply $Spec \Rightarrow \Box P$:
1. By the Invariance Rule, with the property (iii) as hypothesis we obtain that $I \wedge \Box[Next]_{var} \Rightarrow \Box I$.
2. By the Imply Generalization Rule, we obtain that $\Box I \Rightarrow \Box P$.
3. By the above two formulas, it results that $Init \Rightarrow P$ and $\wedge\Box[Next]_{var} \Rightarrow \Box P$. The progress condition V, which describes only what must eventually happen, is not needed for proving safety properties. So, we have proved that $Spec \Rightarrow P$. All we have to do is to find an invariant that fulfills the properties (i)-(iii). We define the invariant I by $I == prodWait + consWait + \Upsilon\left(pc_{prod} \in \{7,10\}\right) + \Upsilon\left(pc_{cons} \in \{11,16\}\right) = 1$, where $\Upsilon : \{true, false\} \rightarrow \{0,1\}$ such that $\Upsilon(true) = 1$ and $\Upsilon(false) = 0$. The proofs of (i) and (ii) are immediate. For the third property we give in the next table the cases in which the predicate I is true:

	prodWait	prodCons	$\Upsilon(pc_{prod} \in \{7,10\})$	$\Upsilon(pc_{cons} \in \{11,16\})$
case1	1	0	0	0
case2	0	1	0	0
case3	0	0	1	0
case4	0	0	0	1

In the first case, only the action b_2 can take place, because it is the only one that verifies the conditions from the predicate I. The execution of action b_2 determines the the following changes of the variables: $prodWait' = 0$, $b'[last] = c$, $last' = last + 1$, $pc'_{prod} = 10$. The other variables, $consWait$ and pc_{cons}, remain unchanged. From these relations it results that $I' = prodWait' + consWait' + \Upsilon\left(pc'_{prod} \in \{7,10\}\right) + \Upsilon(pc'_{cons} \in \{11,16\}) = 0+0+1+0 = 1$, that is $I \wedge b_2 \Rightarrow I'$.
The other implications are proved in the same way for all the other cases and actions from $Next$.■
The demonstration of other safety properties follows the same pattern on which we have used in the proof of previous proposition.

4.2 Proving Liveness Properties

We illustrate the proof of liveness properties by showing that our AOP program, under the weak fairness assumption, will execute a *get* action. Formally, this property is expressed and proved by the following proposition.

Proposition 2. *The AOP program of the Producer-Consumer problem has the property* $(b[0] \neq 0 \wedge last \in (0, Len(b))) \longmapsto (b[last] = 0)$, *which asserts that if* $b[0] \neq 0$ *and* $last \in (0, Len(b))$, *then at some later time* $b[last]$ *will be equal with* 0.

Proof. With the notations $P \stackrel{not}{=} b[0] \neq 0 \wedge last \in (0, Len(b))$ and $Q \stackrel{not}{=} b[last] = 0$ we have to prove that

$$Spec \Rightarrow (P \longmapsto Q) \tag{1}$$

The formal proof of (1) involves three steps:
S1. $Spec \Rightarrow P \longmapsto ((pc_{cons} = 11 \wedge P) \vee (pc_{cons} = 54 \wedge P)$
S2. $Spec \Rightarrow (pc_{cons} = 11 \wedge P) \longmapsto (pc_{cons} = 16 \wedge Q)$
S3. $Spec \Rightarrow (pc_{cons} = 54 \wedge P) \longmapsto (pc_{cons} = 16 \wedge Q)$
The proof of S1. Control in the consumer thread is at either 11, 46, 48 or 54. The locations 46 and 48 of control depend on the truth value of the conditions: $last = 0$ and $prodWait > 0$. Therefore, the only values of control which will interest us are 11 and 54.
The proof of S2. We will verify the three hypotheses of WF rule.
S2.1. We have to prove that $(pc_{cons} = 11 \wedge P \wedge Next_2) \Rightarrow Q'$, where $Next_2 = get \vee c_2$.
We have two cases:
a) $pc_{cons} = 11 \wedge b[0] \neq 0 \wedge last \in (0, Len(b)) \wedge prodWait = 0$. If *get* action takes place then
$b'[i] = b[j]$, $\forall j \in [1, (last-1)] \wedge i \in [0, (last-1)] \wedge last' = last - 1 \wedge pc'_{cons} = 16$. These predicates imply that $b'[last'] = 0 \wedge pc'_{cons} = 16$ i.e.Q'.
b) $pc_{cons} = 11 \wedge b[0] \neq 0 \wedge last \in (0, Len(b)) \wedge prodWait \neq 0$. After that *get* action took place, we obtain $b[i] = b[j]$, $\forall j \in [1, (last-1)] \wedge i \in [0, (last-1)] \wedge last' = last - 1 \wedge pc'_{prod} = 34 \wedge pc'_{cons} = 16$. Last predicate implies that $b'[last'] = 0 \wedge pc'_{cons} = 16$ i.e. Q'.
S2.2. We will prove the second hypothesis of WF rule, i.e. $(pc_{cons} = 11 \wedge P \wedge Next \wedge \neg Next_2) \Rightarrow (P' \vee Q')$. From the definition of $Next$, the previous implication becomes $(pc_{cons} = 11 \wedge P \wedge Next_1) \Rightarrow (P' \vee Q')$.
We have three cases:
a) $(pc_{cons} = 11 \wedge P \wedge put) \Rightarrow (b'[last] = c \wedge last' = last + 1 \wedge pc'_{cons} = 16 \wedge \forall i \in (last'+1, Len(b)-1)$:
UNCHANGED$b[i]) \Rightarrow (pc'_{cons} = 16 \wedge b'[last'] = 0) = (pc'_{cons} = 16 \wedge Q')$.
b) $(pc_{cons} = 11 \wedge P \wedge b[last] = 0 \wedge b_2) \Rightarrow (prodWait' = prodWait - 1 \wedge b'[last] = c \wedge last' = last + 1 \wedge pc'_{cons} = 16 \wedge \forall i \in (last'+1, Len(b)-1)$: UNCHANGED$b[i]) \Rightarrow (pc'_{cons} = 16 \wedge b'[last'] = 0) = (pc'_{cons} = 16 \wedge Q')$.
c) $(pc_{cons} = 11 \wedge P \wedge b[last] \neq 0 \wedge b_2) \Rightarrow (prodWait' = prodWait - 1 \wedge b'[last] = c \wedge last' = last + 1 \wedge consWait' = consWait - 1 \wedge b'[i] = b[j]\ \forall j \in (last-1) \wedge i \in (0, last-2) \wedge last' = last - 1 \wedge pc'_{cons} = 16) \Rightarrow (pc'_{cons} = 16 \wedge b'[last'] = 0) = (pc'_{cons} = 16 \wedge Q')$.
S2.3. The prove of the third hypothesis $(pc_{cons} = 11 \wedge P) \Rightarrow ((pc_{cons} = 11 \vee pc_{cons} = 54) \wedge b[0] \neq 0)$ is immediate. Finally, applying the SW rule with the implication $SF(c_2) \Rightarrow WF(c_2)$ we obtain that $\Box[Next]_{var} \wedge WF(get) \wedge WF(c_2) \Rightarrow (P \longmapsto Q)$, i.e. the formula from S2. Analogously, the step S3 can be proved.■

5 Conclusions

This paper presents some results that were obtained during our research concerning concurrency and its relationship with aspect-oriented approaches. This research drove us to analyze and prove the correctness of concurrent AOP programs written in AspectJ. In this context we had as objectives to: (i) specify the behavior of concurrent AOP programs, and (ii) demonstrate that the properties used to prove the correctness of concurrent programs are preserved for concurrent AOP programs. In this paper, we have presented the proofs of only two properties namely, the mutual exclusion of control threads in concurrent AOP programs and, a special case of liveness property. The later property partially resolves the conditioned synchronization problem of processes. For our proofs we used an aspect-oriented solution of the Producer-Consumer problem.

References

[1] R. Douence, O. Motelet, and M. Sudholt, "A Formal Definition of Crosscuts", Technical Report no 01/3/INFO, 2001.

[2] The AspectJ Team, *The AspectJ(TM) Programming Guide*, 2003.

[3] E. A. Emerson, *Temporal and Modal Logic*, Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, 1995.

[4] R. Jagadeesan, A. Jeffrey, and J. Riely, "An Untyped Calculus of Aspect-Oriented Programs", *European Conference on Object-Oriented Programming*, 2003.

[5] R. Laddad, *AspectJ in Action. Practical Aspect-Oriented Programming*, Manning Publications, 2003.

[6] L. Lamport, "An Axiomatic Semantics of Concurrent Programming Languages", *Lecture Notes of the Advanced Seminar on Logics and Models for Verification and Specification of Concurrent Systems*, France, 1984.

[7] L. Lamport, *Specifying Systems*, Preliminary Draft, 2002.

[8] Z. Manna, A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems Specification*, Springer-Verlag, 1992.

[9] S. Owicki, L. Lamport, "Proving Liveness Properties of Concurrent Programs", *ACM Transactions on Programming Languages and Systems*, Vol. 4, 1982.

[10] D. Walker, S. Zdancewic, and J. Ligatti, "A Theory of Aspects", *International Conference on Functional Programming*, 2003.

```
1.public class Buffer{
2. private char[] b;
3. private int last;
4. public Buffer(int dim){
5.  b=new char[dim];
6. }
7. public synchronized void put(char c)
8. {
9.  b[last++]=c;
10.}
11.public synchronized char get()
12.{
13. char c=b[0];
14. System.arraycopy(b, 1, b, 0, last-1);
15. return c;
16.}
17.public int getLast(){
18. return last;
19.}
20.public int getLength(){
21. return b.length;
22.}
23.}
24.public abstract aspect Concurrency{
25.protected int prodWait, consWait;
26.pointcut bufferPut(Buffer b):call(public void put(char))&&target(b);
27.pointcut bufferGet(Buffer b):call(public char get())&&target(b);
28.}
29.public aspect CondSinPut extends Concurrency{
30. before(Buffer b): bufferPut(b){
31.  if (b.getLast()==b.getLength()){
32.   try{prodWait++;
33.    wait();}
34.   catch(InterruptedException e){e.printStackTrace());}
35.  finally{prodWait--;}
36.  }
37. }
38. after(Buffer b): bufferPut(b){
39.  if(consWait>0)
40.     notifyAll();
41. }
42.}
43.public aspect CondSinGet extends Concurrency{
44. before(Buffer b): bufferGet(b){
45.  if (b.getLast()==0){
46.   try{consWait++;
47.    wait();}
48.   catch(InterruptedException e){e.printstackTrace();}
49.   finally{consWait--;}
50.  }
51. }
52. after(Buffer b): bufferGet(b){
53.  if(prodWait>0)
54.     notifyAll();
55.}
```

Figure 1: AspectJ program for the PC problem

module *Buffer*

import *Nat*

parameters

$b, last, pc_{prod}, pc_{cons}$: VARIABLE

Data: CONSTANT

assertions $0 \notin Data$

predicates

$Init \triangleq \land \forall i \in 0..(Len(b)-1) : b[i] = 0$
$\land last = 0$
$\land pc_{prod} = 0$
$\land pc_{cons} = 0$

actions

$a_1 \triangleq \land pc_{prod} = 7$
$\land c \in Data$
$\land pc'_{prod} = 32 \lor a_2$
$\land before_1$
$\land$UNCHANGED $< b, last, pc_{cons} >$

$a_2 \triangleq \land b[last] = 0$
$\land b'[last] = c$
$\land last' = last + 1$
$\land pc'_{prod} = 40 \lor pc'_{prod} = 10$
$\land after_1$
$\land \forall i \in (last'+1)..(Len(b)-1) :$ UNCHANGED$b[i]$
$\land$UNCHANGEDpc_{cons}

$put(c) \triangleq a_1$

$a_3 \triangleq \land pc_{cons} = 11$
$\land pc'_{cons} = 46 \lor a_4$
$\land before_2$
$\land$UNCHANGED $< b, last, pc_{prod} >$

$a_4 \triangleq \land b[0] \neq 0$
$\land \forall j \in 1..(last-1) \land i \in 0..(last-2) : b'[i] = b[j]$
$\land last' = last - 1$
$\land pc'_{cons} = 54 \lor pc'_{cons} = 16$
$\land after_2$
$\land$UNCHANGED $< pc_{prod} >$

$get \triangleq a_3$

$getLast \triangleq$ UNCHANGED $< b, last >$

$getLength \triangleq$ UNCHANGED $< b, last >$

$v \triangleq < b, last, pc_{prod}, pc_{cons} >$

temporal

$Spec_{buffer} \triangleq \land Init$
$\land \Box[put(c) \lor get]_v$
$\land WF_b(get)$

Figure 2: TLA specification of Buffer

module *Concurrency*

import *Nat*

parameters

$b, prodWait, consWait, pc_{prod}, pc_{cons}$: VARIABLE

predicates

$Init_1 \triangleq \land b \in Buffer$
$\land prodWait = 0$
$\land consWait = 0$
$\land pc_{prod} = 0$
$\land pc_{cons} = 0$

temporal

$Spec_{con} \triangleq Init_1$

Figure 3: TLA specification of Concurrency

module *CondSinPut*

actions

$b_1 \triangleq \land pc_{prod} = 32$
$\land prodWait' = prodWait + 1$
$\land$UNCHANGED $< b, consWait, pc_{prod}, pc_{cons} >$

$b_2 \triangleq \land pc_{prod} = 34$
$\land prodWait' = prodWait - 1$
$\land a_2$
$\land$UNCHANGED $< b, consWait, pc_{cons} >$

$before_1 \triangleq b_1 \lor b_2$

$after_1 \triangleq \land pc_{prod} = 40$
$\land pc'_{cons} = 48$
$\land pc'_{prod} = 10$
$\land$UNCHANGED $< b, prodWait, consWait >$

$v_1 \triangleq < b, prodWait, pc_{prod}, pc_{cons} >$

temporal

$Spec_{condSinPut} \triangleq \land Init_1$
$\land \Box[b_2]_{v_1}$

module *CondSinGet*

actions

$c_1 \triangleq \land pc_{cons} = 46$
$\land consWait' = consWait + 1$
$\land$UNCHANGED $< b, prodWait, pc_{prod} >$

$c_2 \triangleq \land pc_{cons} = 48$
$\land consWait' = consWait - 1$
$\land a_4$
$\land$UNCHANGED $< b, prodWait, pc_{prod} >$

$before_2 \triangleq c_1 \lor c_2$

$after_2 \triangleq pc_{cons} = 54$
$\land pc'_{prod} = 34$
$\land pc'_{cons} = 16$
$\land$UNCHANGED $< b, prodWait, consWait >$

$v_2 \triangleq < b, consWait, pc_{prod}, pc_{cons} >$

temporal

$Spec_{condSinGet} \triangleq Init_1$
$\land \Box[c_2]_{v_2}$

Figure 4: TLA specifications of CondSinPut and CondSinGet

Crenguţa Mădălina Bogdan
"Ovidius" University Constanta
Computer Science and Numerical Methods
Address: Mamaia Blvd., 124, 900527
E-mail: cbogdan@univ-ovidius.ro

Luca Dan Şerbănaţi
"Politehnica" University Bucharest
Department of Engineering in Foreign Languages
Address: Splaiul Independentei, 313, 77206
E-mail: luca@serbanati.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 100-104

New aspects of Software Development in Economy

Cornelia Botezatu, Cezar Botezatu

Abstract: As competition has increased, whereas the issues that trade activities are now facing have become more complex and unprecedented problematic - the design and development of software products call for new requirements and performances to match them.
The article undertakes to outline a part of these requirements, which are of interest to both software system programmers and their end users, regarding the analysis, patterning, design and implementation of IT applications.

Keywords: e-activity, virtual enterprise, e-manufacturing

1 Introduction

The last decade produced major changes in ITC, allowing profound and irreversible transformations in the entire society (economy, learning, spending time,administration etc.), contributing to new forms of habits, products andservices appearance. The technological mutations produced massive changes in people and organizations practices, transforming the communication, working, learning ways and making compulsory the organization rethinking, to practical and coherent methods and rules. The digital era offers new business models. Their success consists in getting the needed information, its interpretation and the possibility of real time transactions. So far, the companies could not entirely capitalize the advantages of the electronic systems for businesses. Although the exchange of information and the negotiations were taking place through Internet, a critical moment was reached when the documents had to be printed and signed manually and then had to be sent to the destination. We are now part of a society transformation, from a hard copy based information to one based on the electronic form. Not one human activity can stay away of this informational revolution. The information society is developing everywhere:

- In companies and organizations,especially in the conception-production processes (groupware, workflow.)
- In distribution and commerce,developing electronic commerce
- In education, converging towards the new virtual universities and all new teaching methods, based on ITC
- In administration, allowing free access to public information and the perspective of virtual administration
- In editing sector (online magazines, online libraries, electronic books, etc.)

2 New aspects of software development

The fierce competition on the sales market creates special adaptation and management difficulties for the production and services companies, in order to find new ways of increasing their shares of this market. Through the software developers it is easy to establish, to build and to maintain product selling virtual relationship and it is relatively easy to facilitate the company penetration on virtual market, using a web interface or an integrated management system. On the other side, service selling virtual markets allow more possibilities, allow the creation of more sophisticated commercial dependencies, and reunite different types of companies, meaning more unified business procedures. The electronic commerce represents a new technology, which offers numerous options and models for making business using Internet, such as:

- E-shop;
- E-mall;
- E-procurement;
- E-auction;
- Virtual community;

- E-service providing;
- Information brokerage;
- Publicity models.

Current complex ways of living, learning, travelling, administrating or governing require a deep understanding, knowledge and experience from managers, who have to decide what can be done with information systems, in order to obtain the foresighted benefits. There are series of economical concepts which can explain how information system can be used in present, in order to improve the enterprises. We live in a demands era. The customers want, from software developers, more and more: they want high quality for offered products, fast updates, customized services, integrated systems and low-cost delivery, and all of these as fast as possible. On the other hand, the software developers must understand all these requirements, to be up to date with the last achievements in ICT field, to propose more efficient solutions for all issues, to design integrated solutions, to ensure data security, confidentiality and trust in electronic transactions. As a result, we identify for design demand, at least two directions:

- First regarding design methodologies and development techniques used in every stage of the life cycle of an information system. Now are commonly used CASE instruments, or development templates.
- Second regarding new information and communication technologies, hardware and software platforms that influence directly both the nature of proposed technical solutions for new applications and their number. Can be mentioned here new ways of application development,like client-server, distributed and web technologies.

3 Current technologies and requirements for development

3.1. CASE instruments. In order to respond to the new demands related to the complexity of the questions asked and low amount of time allowed for solving them, the software developers use more and more the CASE instruments, in different stages of development and production of information systems. The continuous and major changes regarding design and development of software products increase the number of new software products, that aim to assist and guide the human factor through the design and development stages of a software product. The list of CASE instruments is large, starting with those used exclusively for diagrams, and ending with integrated software platforms, with CASE instruments incorporated for assistance in all stages of the project. Thus, the CASE instrument concept covers now at least the following components: diagrams editor, repository, browser, code generator, reverse-engineering instruments, XML parser, and documentation generator. Some of UML CASE instruments used by informational system designers were: Rational Rose, S-CASE, Ration CRC, Together, Poseidon and Rhapsody.

Designer 2000, produced by Oracle, is a product dedicated to system analysts, programmers and project managers. It contains a powerful integrated set of CASE (Computer Aided Software Engineering) instruments, having as main objective to assist the human factor into software design and development stages.

Microsoft Visio Professional 2002 supply diagrams type solutions for documenting and communicating of large ideas, information and systems pallet.

Microsoft Office Visio 2003 introduce a new set of diagrams and schemas, making from this product an "erudite" dedicated to visual documentation development specific to various fields. In addition, the software, web sites and DB developers and network administrators have on disposal, using specialized diagrams offered by Visio, a bright assistant dedicates to automation of specific activities.

IBM Rational Unified Process (RUP) is a software paradigm, designed, developed and maintained as a software product. RUP created on UML base, delivered on-line using Web technology.

A modern product, **JBuilder 6** includes a series of improving facilities for assisting the participant for all project period.

3.2.Design Patterns. Designing software is a difficult process, and the object oriented design - often used lately - of reusable systems is even harder. The solution must be specific, but also general enough to be applied in the future, in order to avoid "reinventing the wheel" each time (or at least to minimize this possibility).

Often enough, an inexperienced designer is overwhelmed by the number of available options and has the tendency to revert to non-object techniques used in the past. An experienced designer knows that he doesn't have to solve every issue starting from scratch, but by reusing solutions from previous projects. Once he has discovered a good method, he will always use that method. This kind of experience is only a part of what makes an expert out of a designer, and this method is a template for him.

Templates like ACE, MicrosoftMFC and DCOM, JavaSoft RMI and OMG CORBA implementations have an increasingly important role in current software development. The main benefits of using the templates in object oriented applications are coming from the modularity, possibility of reuse, extensibility and reverse engineering offered to the developers.

3.3. New software architectures. The design and development of e-commerce that must allow operating with maximal security conditions for all transaction participants, operating with information, promotion and other classic commercial actions, with maximum efficiency, are all tasks of the designer.

The information security becomes highly important for safety assurance, protection and authenticity of the computer memorized or transmitted data. Lately in developed countries, the paper became just a tool for presenting information, not for archiving or transporting it. Computers and computer networks assumed these functions. There were sought and found solutions for replacing seals, stamps and hand-made signatures from classic documents with their electronic alternatives, based on classic cryptography and public keys.

On the virtual market, both large companies and the small enterprises start with equal chances. The high discrepancy that advantages one or another on the real market is gone, and advantages like those coming from better geographical/strategical positioning of the business become irrelevant. The most important factors to be considered in deciding who is going to be the winner are time and knowledge.

The client server architecture, which is “de facto” for the majority of actual information systems, is offering a large accessibility to DBMS information resources.

The client-server DBMS form the foundation for new solutions, covering financial and economical activities,such as BI (Business Intelligence), ERP (Enterprise Resource Planning), CRM(Customer Relationship Management) or SCM (Supply Chain Management).

The platform architecture is the physical implementation of application (figure 1) a have the following tasks:

- To implement client-server type information architecture having Internet and WWW technologies as support;
- To assure data and information management by including of Web server, terminal server and application server functions and encoding of numerous Java applet to transfer the client request to the receiver.
- To simulate a central server,accessed using Web interface with CGI (Common Gateway Interface) and Java applets;
- To be flexible, scalable and to assure components interconnectivity;
- To platform independent;

The browser design for a better interface with system server imposes the usage of object oriented programming languages and new techniques such as serialization, RMI (Remote Method Invocation) or CORBA (Common Object Request Broker Architecture).

The object oriented information technologies and the software methods combined in distributed on Internet offer to the companies, private persons, governments an infrastructure, which allow to:

- To create a virtual market of goods and services: Consumer to Consumer-C2C, Consumer to Business-C2B, and Business to Government-B2G.
- Information exchange (Governmentto Business-G2B, Government to Consumer-G2C)
- To improve the payment system for the state taxes by lowering the costs (Consumer to Government-C2G)
- To develop the after-sales services and marketing directly consumer-oriented
- E-Commerce (Business to Business, Business to Consumer)

Among all these applications, the leader is represented by the Business to Business (B2B) transactions. From a ŞBusiness to BusinessŤ customer’s point of view the information management belonging to the buyer must be formed on the buyer’s site to integrate with the other information. Likewise, the information belonging to the buyer must be stored in his server in order to allow e-payments, the working flow, and the answer by Intranet. A new option in e-commerce is the Business-to-Employee transaction, which refers to the transactions inside the company involving the employed staff of the company and made by its own Intranet system.

When starting such a business concerning e-beneficiary, the usual problems might be: who the clients might be, what their desires are, where they are placed on Intranet, what the company and product strategies are, what their

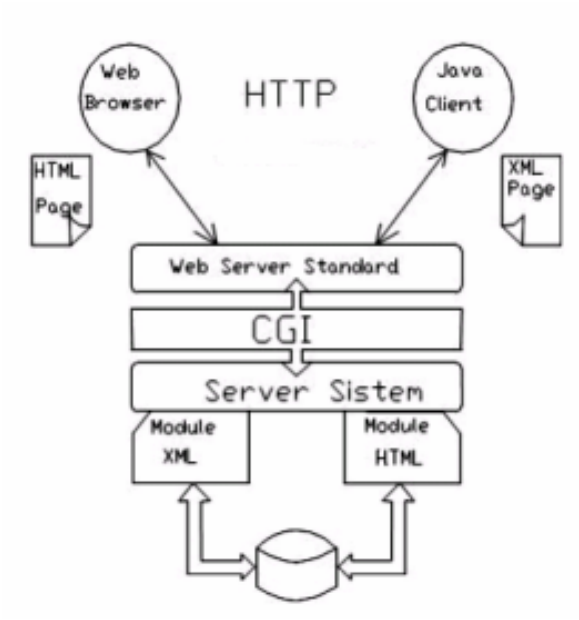


Figure 1: Client-server architecture

particular characteristics are, what ways of selling can be chosen, what investment is necessary and sustainable by the company.

3.3. Other requirements regarding the design of a commercial site. The company that designs a software application for creating a virtual selling environment should now take into account a series of problems that may influence directly the suggested solution, among which:

- What way is chosen to connect the server to the Internet
- Who makes the page design in order to display the products professionally
- Who is going to maintain this site
- What command options/facilities will be accepted
- How the delivery of goods is going to be made
- What payment methods will the clients use
- What informing and reporting situations are necessary regarding the transactions already done
- Who and how the access to the company's data, site and soft will be made
- What procedures are there to check the data
- What technical equipment is required to answer the infrastructure of the application

Concerning the layout of Intranet pages used in e-commerce it is forbidden to overload the design: an overloaded graphics with animation and lots of music could impress us for the moment, but most users do not have time to wait and see on the screen the company's fancy text, they cannot focus on the important messages, they are annoyed thus giving up to read that information. That is why many Internet pages contain this much information they are over-loaded with options and navigation ways. The resultis practically a commercial labyrinth which will cause the loss of clients.

The well made Internet pages are those that manage to set a balance between an attractive graphics, a pleasant one and the loading speed of the pages on screen. This should be easy to access and the searched information by every client is easily placed, not depending on the navigation way chosen by him. That's why we mention that the page layout has a great importance.

E-commerce offers new and intelligent solutions to quantify the impact upon its clients such as: online answer options to polls, options to participate in different competitions or member subscriptions, which allow them to get information about clients although they remain anonymous. Most commercial Internet pages have interactive ways to measure the impact upon clients, but they can be used in marketing activities such as: periodical news, market studies, the marketing of products etc.

4 Summary and Conclusions

The choice of one of these solutions is determined, in a great measure, by performance criteria and business planning for each company for e-commerce activity expected results. Another success factor represent the experience, professional training, innovative spirit, creativity, adapting and using existent solution capacity accordingly with real market design of design team.

Taking a look around, it is a sure thing that different forms of e-commerce development will contribute to exploration and increasing the globalization process of economy, and specially, the commerce globalization. Such a thing will conduct to a coherent universal functioning system in the new environment, which will cover all aspects of society: economic, legal, financial, data security and safety and customer protection. All these will be considered as particular task in software developers' job, which will use or create new ways of assisting design and development activities regarding informational systems, of standardization or reuse of the existing software tools and modules.

References

[1] User-to-Business Patterns using WebSphere Enterprise Edition, *Patterns for e-Business Series*, IBM Redbook SG245161.

[2] E-Commerce Patterns using WebSphere Commerce Suite, *Patterns for e-Business Series*, IBM Redbook SG245165.

[3] Stephen C. Glazier, *Patent Strategies for Software, e-Commerce, the Internet, Telecom Services, Financial Services, and Business Methods*, BI Law and Business Institute, ISBN 0966143779.

[4] Janica Reynolds, *The Complete E-Commerce Book: Design, Build and Maintain a Succesfull Web-based Business*, CMP Books, ISBN 157820061X.

[5] Michael J. Cunningham, *B2B: How to Build a Profitable E-Commerce Strategy*, Perseus Pr, ISBN 0738203343.

[6] C. Botezatu, *Modern tendencies regarding design of informational systems*, Universul Juridic.

[7] C. Botezatu, "Current elements regarding the development of sowftware products" *Revista Informatica Economica*, no.1(29)/2004, pp.91-95, ASE Bucharest, Ed. INFOREC, ISSN1433-305.

[8] SIMPLIFICATION.COM, *Electronic business and the simplification of administration*, edited under the patronage of the european Economic Commission of United Nations.

[9] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Paterns*, 0254.

[10] http://www.postgresql.org/; http://www.intraweb.ro/txt/

Cornelia Botezatu
Romanian-American University
IT Management Systems Department
Address: B, EXPOZITIEI Avenue, District 1, Bucharest
E-mail: c2botezatu@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 105-112

Architecting J2EE based Applications on Multiple Layers

Cristian Butincu, Mitică Craus, Dan Gâlea

Abstract: This paper presents a layered model that can be used to construct well designed enterprise applications. The proposed model describes how an enterprise application can be divided into 9 essential layers as follows: Persistence, Domain, Services, Facades, Factories, DTO, Delegates, Application and Presentation. This model encapsulates a clear separation of the roles of each layer. The main goal of the model is to provide all the necessary information that the enterprise application designers need in order to build an enterprise application backbone that will support the development of efficient, scalable and maintainable enterprise applications. Another goal is to isolate the end programmers from the complexity of enterprise design patterns.

Keywords: enterprise application, architecture, design, multiple layers, role separation

1 Introduction

J2EE design patterns [2], [3], [6] are vital for anyone building J2EE applications. Ranging from basic to complex patterns they answer important questions as how to use and access EJBs correctly and how one should design the data flow between the components of an enterprise application. Well designed EJB applications make use of design patterns. The main benefit of using design patterns is that they improve the quality and robustness of the entire application, provide code reusability and a clear and relatively easy to understand design. Unfortunately, well designed enterprise applications are not so easy to be developed, mainly because there is still a deficit of information about how to design a good system. Enterprise application developers often make costly design mistakes every day [7]. Moreover, learning good design is particularly difficult for new enterprise application programmers, many of whom have never built distributed systems before and don't understand the fundamental needs that influence distributed systems design. The key of making a good J2EE design is to abstract parts of the application and to reduce at maximum the dependencies between application modules. This goal can be achieved if the enterprise application is divided into several layers. This paper details how to design an enterprise application by using a model made up by 9 layers. Each layer is composed of multiple classes, or components pieced together. By building the system using layers, localization of data and behavior allows for decreased dependency between classes and objects, providing a more robust and maintainable design. A similar model based on 5 layers is discussed in [6]. However, the model proposed in this paper introduces additional layers and provides a better separation of roles among layers.

2 The use-case-driven approach

One of the most effective approaches taken in order to design an enterprise application [8] is the use-case-driven approach [6]. The first step is to identify the set of use cases that the system will need to support. These use cases are very important because they can influence the design decisions of the enterprise application. The second step is to identify what are the semantics and user interactions of each use case. Once these specifications are set, the next step is to analyze the use case model. Based on the result of this analysis, enterprise application designers will have enough information to begin to sketch the concrete design of the final enterprise application. The layered model presented in this paper fully supports the use-case-driven approach.

3 The domain model and the domain layer

The domain model comprises all the objects upon which actions are performed. Using the use-case-driven approach, the domain model can easily be derived from the identified use cases. Because the initial domain model granularity can be too coarse or, on the contrary too grained, in order to provide superior enterprise application performances, the domain model can be further refined in several steps until the desired domain granularity is reached. This operation is typically carried out by domain experts. One of the 9 layers of the model presented

in this paper is the domain layer. In general, the domain layer of an enterprise application is composed of entity beans. These entity beans can be either Container-Managed Persistence entity beans (CMP entity beans) or Bean-Managed Persistence entity beans (BMP entity beans). The domain layer is the layer in which domain model resides.

3.1 Bypassing the domain layer

There are situations where the domain model, and therefore the domain layer do not exist at all. In this kind of situations, session beans bypass the domain layer and make direct calls into the persistence layer, a low level layer that directly accesses underlying data bases. Except some exceptional cases that will be detailed at the end of this section, this approach should never occur in an enterprise application design. Some of the enterprise application designers choose this approach for one of the following reasons:

- Quick build of prototypes - the enterprise application is not intended to have a long life span, or its domain model will change often over time
- Trivial domain models - the enterprise application domain model is very simple and the time to deliver the solution is very short
- Performance reasons - used when implementing a use-case that is by default read-only

The last reason is one of the aforementioned exceptional cases in which the domain model should be bypassed. However, even in this case, the domain model, and therefore the domain layer should still exist in order to provide an OO base for the other use cases. As a conclusion, the domain layer should always be present in an enterprise application design although in some exceptional cases it should be bypassed for performance reasons.

4 Enterprise application layers

Using the layered model approach, an enterprise application is being designed layer by layer. Design patterns [5], [9] play the most important role in each layer design. They can dramatically influence the inside architecture of each layer. The 9 layer model presented in this paper is illustrated in Figure 1. Figure 1 emphasizes the interactions and dependencies of the layers in the enterprise application. The DTO (Data Transfer Object) Layer establishes three direct dependencies: implementations of Application Layer, Facades Layer and Factories Layer depend on implementation of DTO Layer.

The application structure design depicted in Figure 1 can be viewed as a topological directed graph that contains the enterprise application layers. All the data managed by the enterprise application flows through these layers in both directions, from the client side layers to the server side layers, one layer at a time. The adjacent layers are connected by edges that include assembly connectors. In this case, an assembly connector is a connector between two layers that defines that one layer provides the services that the adjacent layer requires. After the enterprise application is fully developed [4], [10], the implementation of each layer can vary independently without triggering modifications of the adjacent layers, provided that the general contract of supported interfaces is not altered. The role of each layer is described in the next sections.

4.1 Classification of enterprise application layers

The enterprise application layers can be classified in server side layers and client side layers as follows:

- Server side layers: Persistence Layer, Domain Layer, Services Layer, Facades Layer, Factories Layer
- Client side layers: Delegates Layer, Application Layer, Presentation Layer
- Common layers: DTO Layer

4.2 Role of enterprise application layers

Persistence layer

This layer contains all the logic required to make the domain model persistent in a data store. The Persistence Layer provides the RDBMS abstraction, that is, any database can be plugged into the enterprise application without the need to modify any layers that depend on it.

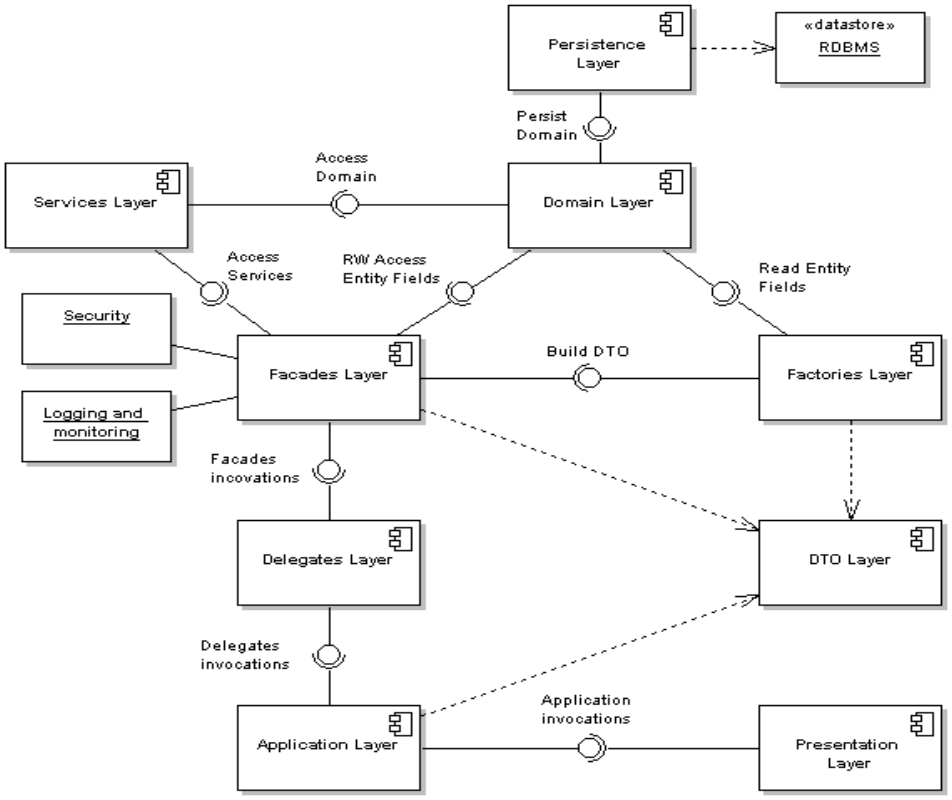


Figure 1: Enterprise application layers

Domain layer

This layer contains all the objects that came out of the object-oriented analysis of the business problem. The Domain Layer is the layer in which domain model resides. The Facades Layer, Services Layer and Factories Layer delegate many of the requests to this layer. A well designed Domain Layer is often application independent and can be reused across applications.

Services layer

This layer contains all the session beans of the enterprise application that resulted from the analysis phase. These session beans can be either Stateless Session Beans or Stateful Session Beans. The Services Layer is the layer that Facades Layer calls in order to invoke business logic specific to particular use cases (typically on multiple domain objects and/or other services). It is responsible to control the transactions that the use cases run under, and to handle any delegation and workflow logic between domain objects required to fulfill a use case.

Facades layer

This layer decouples the server side layers from the client side layers. All accesses from the client side layers into the server side layers are tunneled through this layer. The Facades Layer extends the concept of classical Session Facade design pattern to wrap both Domain Layer and Services Layer. Bulk update and bulk data retrieve services that work with DTOs are implemented at this layer level. The Facades Layer is a very important layer, since it controls all accesses into the server side layers. Security checks and logging and monitoring mechanisms are also implemented at this layer level. In order to hide Domain Layer implementation details from client side layers, Facades Layer relies on the services that Factories Layer provides.

Factories layer

This layer decouples the Domain Layer from the client side layers by the means of DTO factories (Data Transfer Object factories). The Factories Layer is responsible of building domain DTOs and custom DTOs, both of which reside in DTO Layer. This layer hides the implementation details of Domain Layer from the client side layers.

DTO layer

This layer encapsulates the data transfer objects used as containers of data in both ways, from clients to server and from server to clients. The DTO Layer sits between client side and server side layers. Its main role is to decouple Domain Layer from client side layers. In a secure environment, DTOs encapsulate an additional security rights payload that is used to control access to their contained data.

Delegates layer

This layer abstracts the server side layers from the client side layers. The Delegates Layer hides the EJB API complexity by encapsulating code required to discover, delegate and transparently recover from invocations into server side layers. By the means of this layer, client side programmers are completely decoupled from the technology used on the server side. Therefore, the design and implementation of the server side can vary independently of the client side. A client side logging mechanism is appropriate to be implemented at this layer level. The Delegates Layer is most useful in large projects where client side layers and server side layers are being developed by different teams of programmers. In the early stages of enterprise application development, when the server side part of the application is not available, this layer can provide dummy data to the client side application programmers. This way, the latter ones don't have to wait for the server side programmers to finish their work, and the application development process is accelerated since the work on both client side and server side layers is performed in parallel.

Application layer

This layer is the heart of the client application and controls the workflow between components on Presentation Layer and Delegates Layer. The Application Layer is responsible for managing client-side state, performs syntactic validation on user input, implements application caches and delegates calls to Delegates Layer.

Presentation layer layer

This layer contains all the UI components and their corresponding action listeners. For a stand alone client application, these UI components are Swing or AWT components; for a web based application these are web components such as HTML, JSP or Flash. The action listeners contain callback methods that UI components will invoke when their state changes. The action listeners' role is to make calls into the Application Layer, based on the state of UI components, and to update UI components to reflect changes that occurred during these calls.

5 Development order of enterprise application layers

Based on the layered model described in this paper, there is a natural order in which application layers can be developed. This natural order can be easily inferred from the proposed enterprise application structure design depicted in Figure 1. If we consider the topological directed graph that contains the enterprise application layers, the natural order of layers' development is given by applying a topological sorting algorithm on this graph. Based on the results of topological sorting, DTO Layer and Persistence Layer should be developed first. However there are certain situations when the Persistence layer will not be developed at all by the enterprise application programmers. This happens when the server side developers choose to use a generated persistence logic technology like CMP entity beans, JDO or Object / Relational mapping tools. In this case, Persistence Layer will still exist, but it will be automatically generated. The next layer to be developed is the Domain Layer. After this step is complete, Services Layer and Factories Layer are the next layers to be developed. These two layers can be developed in parallel since all their dependencies are resolved by now. The Facades Layer is the next layer to be developed, followed by Delegates Layer, Application Layer and finally, Presentation Layer. An interesting point here is that DTO layer and Domain Layer are completely decoupled. Although DTOs in general do reflect objects that reside inside of

Domain Layer, the mapping from DTOs to these objects is not a direct one. DTOs and domain objects can be designed at different granularities. Moreover, DTOs can include additional data that is not reflected by domain objects, such as calculated fields. Similarly, domain objects can contain fields that are not reflected by DTOs, like application sensitive fields. Therefore, a special component decoupling logic that keeps track of all these mappings need to be implemented in order to achieve this kind of abstraction. This decoupling logic component is part of Factories Layer and Facades Layer and it is illustrated in Figure 2.

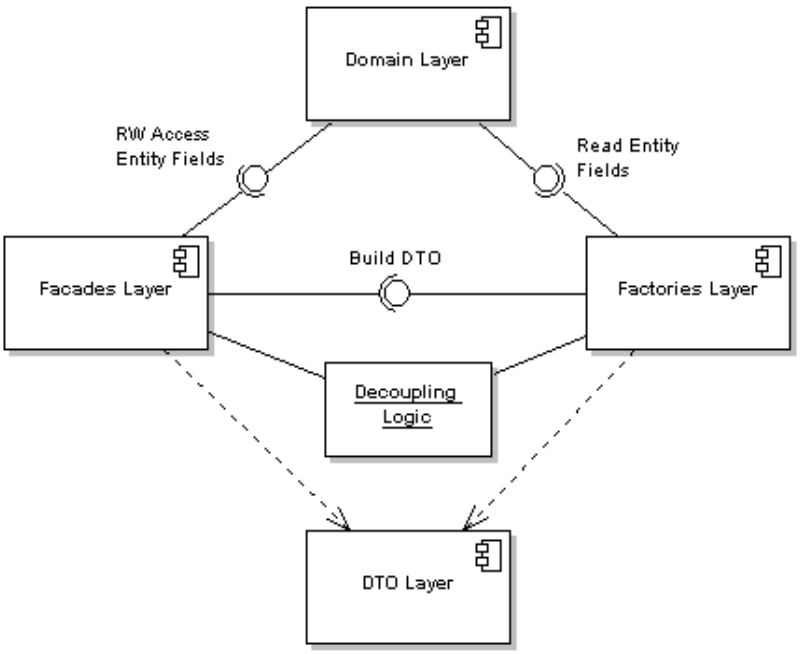


Figure 2: Decoupling logic component

The order of development described here is possible because of the way the whole structure of enterprise application was designed. The key point is that the dependency relationships are unidirectional, and this is the industry best practices. Moreover, enterprise application layers form a directed graph structure with no cycles that results in a topological graph structure.

6 Additional services

Architecting an enterprise application using the layered model presented in this paper provides several other advantages. For example, if the enterprise application has to provide a secure environment, all the needed implementation updates are localized only at the Facades Layer level. In a few words, a Security Service component has to be developed and plugged-in into this layer. All the other layers of the enterprise application will remain unaffected. The process of implementing a security model would trigger major modifications on a poorly designed enterprise application where the majority of components on the server side need to be updated to support security, as compared to the layered model proposed in this paper that will trigger only small modifications localized only at the Facades Layer level. The same holds for a wide range of other services that are global in their nature. Another example of service that can be very easily plugged-in into the system is the Logging and Monitoring Service. As with the aforementioned Security Service, all the updates are localized only at the Facades Layer level. The enterprise application design described in this paper provides the enterprise application programmers with the ability of introducing additional services in a minimum amount of time and with a minimum effort. All of this is possible because of the clear separation of roles that is inherent to this layered model.

7 Implementation example

This section details how a certain use case can be implemented on this architecture. As an example we take into consideration the management of products and manufacturers for a certain company. For this we need at least two tables in the database: one for products and one for manufacturers. The role of each layer for this use case is as follows:

Persistence layer. This layer provides both RDBMS abstraction and read/write access to these two tables. Normally this layer is provided by enterprise application server vendors, although not necessarily.

Domain layer. This layer contains two sets of classes, one for products and one for manufacturers. It provides an object oriented view of product and manufacturer entities. The java classes that need to be developed are:

- ProductEntityBean.java (javax.ejb.EntityBean subclass)
- ProductEntity (javax.ejb.EJBLocalObject sublass)
- ProductEntityHome (javax.ejb.EJBLocalHome subclass)
- ManufacturerEntityBean.java(javax.ejb.EntityBean subclass)
- ManufacturerEntity (javax.ejb.EJBLocalObject sublass)
- ManufacturerEntityHome(javax.ejb.EJBLocalHome subclass)

Note that this model emphasizes the use of local interfaces only for domain.

Services layer. Suppose that the application needs to provide a statistic of most wanted products per regions along with their manufacturers. This kind of information does not relate itself to only a single type of domain entity and therefore such business logic has to be placed in a separate non domain entity that is, inside a service implemented in terms of a session bean. A set o three java classes needs to be developed for this service: the actual session bean class that implements the logic, the local interface that acts like a service front end to the user and the home class that provides the means by which a user can get an instance of this service:

- StatisticServiceBean.java (javax.ejb.SessionBean)
- StatisticServiceLocal.java (javax.ejb.EJBLocalObject)
- StatisticServiceLocalHome.java (javax.ejb.EJBLocalHome)

Note that this model emphasizes the use of local interfaces on services layer as well. All accesses in the server go through facades layer.

Facades layer. Controls all accesses into the server side layers. This layer typically consists of session beans with remote interfaces. Therefore sets of three java classes (bean class, remote interface class and remote home class) need to be provided. To avoid having multiple session beans in this layer, one can develop a generic session bean using the power of runtime generated proxies, a feature that java language offers. Typically this session bean intercepts remote calls, performs security checks, logging and so on and forwards these call inside server side layers, either to Services layer or to Domain layer.

Factories layer. This is where DTO factories (Data Transfer Object factories) reside. For our use case example, this layer needs to provide two DTO factories, that is, two java class files that are responsible to build DTOs for products and manufacturers:

- ProductDTOFactory.java
- ManufacturerDTOFactory.java

These two factories take as input domain entities and they output POJO like objects (DTOs) that are completely decoupled from entity beans or database. To further extend this layer to allow a complete control over DTO granularity please see [1].

DTO layer. This layer consist of DTO java classes. For our example, we need to develop two java classes:

- ProductDTO.java
- ManufacturerDTO.java

In a secure environment, DTO classes can be extended to encapsulate additional security rights payload that is used to control access to sensitive data. This is typically achieved by developing a generic DTO super class that takes care of this special security payload for all DTO subclasses.

Delegates layer. This layer consists of java classes that contain code to discover, delegate (forward calls) and transparently recover from invocations into server side layers. To avoid having multiple classes in this layer, one can develop a generic delegate class using the same approach that can be taken into Facades layer that of java runtime generated proxies. Therefore these two generic implementations, generic delegate and generic facade work hand in hand and greatly reduce the amount of work needed to keep all files in sync.

A client side logging mechanism is appropriate to be implemented at this layer level as well.

Application layer. Consists of java classes that manage client-side state, perform syntactic validation on user input, implement application caches and make calls into Delegates Layer. In our use case example, client side application logic that updates, creates or deletes a product or a manufacturer needs to be developed at this level.

Presentation layer. Contains UI component classes and GUI classes for graphical management of products and manufacturers. Graphical interfaces that display statistics also need to be developed at this level.

8 Summary and Conclusions

The layered model described in this paper contains all the necessary information that the enterprise application designers need in order to build an enterprise application backbone that will support the development of efficient, scalable and maintainable enterprise applications. This model provides a series of key points in the application structure where enterprise application developers can easily plug-in a wide range of additional services, like security services, logging and monitoring services, failover services, caching services and so on. An important layer that is part of this model is the Facades Layer that controls the workflow into the server side layers. This layer extends the concept of classical Session Facade design pattern to wrap both Domain Layer and Services Layer. All accesses from the client side layers into the server side layers are tunneled through this layer. This model also describes a special component decoupling logic whose role is to completely separate the DTO Layer from the Domain Layer. Using the power of runtime generated proxies and custom serialization mechanisms that the Java language offers, some of these layers, specifically DTO Layer, Delegates Layer and Facades Layer can be architected in a generic way [1]. The design structure that this paper presents is intended to add major improvements to the quality and performances of EJB based applications and to speed up the enterprise application development.

References

[1] C. Butincu, M. H. Zaharia, D. Galea, “A new model in design of data transfer layers of distributed applications running on J2EE platform using polymorphic lightweight inter-tier data transfer objects” *Bulletin of Technical University of Iasi*, 2005, in press.

[2] D. Broemmer, *J2EE Best Practices. Java Design Patterns, Automation, and Performance*, Wiley Publishing Inc, 2003.

[3] D. Alur, J. Crupi, D. Malks, *Core J2EE Patterns: Best Practices and Design Strategies, Second Edition*, Sun Microsystems Press, 2003.

[4] E. Roman, R. P. Sriganesh, G. Brose, *Mastering Enterprise JavaBeans, Third Edition*, Wiley Publishing Inc., 2004.

[5] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional Computing Series, 1994.

[6] F. Marinescu, *EJB Design Patterns, Advanced Patterns, Processes and Idioms*, Wiley Computer Publishing, John Wiley and Sons, Inc., 2002.

[7] H. Sheil, *J2EE project dangers*, http://www.javaworld.com/javaworld/jw-03-2001/jw-0330-ten.html

[8] I. Singh, B. Stearns, M. Johnson, *Designing Enterprise Applications with the J2EE Platform, Second Edition* Addison-Wesley, 2002.

[9] J. W. Cooper, *The Design Patterns Java Companion*, Addison-Wesley Design Patterns Series, 1998.

[10] R. Monson-Haefel, *Enterprise JavaBeans, 3rd Edition* O'Reilly, 2001.

Cristian Butincu, Mitică Craus, Dan Gâlea
Technical University "Gh.Asachi"
Department of Computer Science and Engineering
Address: 53 A, Dimitrie Mangeron Street, 700050 Iasi, ROMANIA
E-mail: cbutincu@yahoo.com, craus@cs.tuiasi.ro, galea@cs.tuiasi.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 113-117

New Rules in Business Environment

George Căruţasu, Cornelia Botezatu

Abstract: In the last period, the corporations have developed their activity over the national borders and even the continent, activity being now easier because of ICT support. The political changes in today world have opened new markets for the existing corporation.Thus, differences between countries regarding production costs impose as reliable economical solution, to open local production facilities, by building,merging with, or acquisition of local already present factories. The business environment has suffered a major change, by eliminating the disadvantage of geographical border using IT support, transnational companies being more and present in todayŠs economic life. In addition, the political and social changes around the globe facilitate the new markets opening. In this case, the enterprise must face the new challenges of e-economy, considering the two presented models: extended enterprise and interest group as a free choice affiliation to a dynamic structure. The most effort in those direction target the integration of independent organization in mega structures with more then hundreds of members. The integration has technology and management coordinates.In addition, an important aspect of e-activity or transnational company activity is the existence of a continuous legal frame, who rules the economical-activity in European space.

Keywords: e-activity, virtual enterprise, e-manufacturing

1 Introduction

The *e-economy* might be seen as branch of conventional economy, developed around an information core and it ways of circulation. In this concept, an important underline is the information usage and manipulation in many various ways. Therefore, in the beginning, the IT involved in an enterprise appears as “islands” of processes automation. Because of late IT evolution, the enterprises have the meanings necessary to develop activities in virtual environment, based on virtual processes, where the keywords are *integration, collaboration* and *efficiency.*

The enterprise which evolve in today’ economy must face the globalization process, because of large area market policy and very strong IT influence. VE offer the most reliable organization structure, but it have been combine with companion measures, involving large transition process. Because of organization manner, the enterprise is flexible,giving the possibility of reducing at maximum the time-to-market for new products. In addition, because of extended partnership, VE can manage more precise new situations, which can occur in market, being more *adaptive*. Another critical success factor of these days is the simulation process, once the VE partners are agree about competencies assumption, the real manufacturing can start, saving time and resource usage.The transition process requires a sustained effort, having a 30% rate of success. The case study offer, from models view, the complexity of the process.

In the e-economy, a series of relationships has been shaped, relationships, which bring together the categories mentioned before:

- *business to business (B2B)*, company to company, is the most extended relationship, and it include all electronic transaction between companies;
- *business to customer (B2C)*, it refer to transaction between company and individual customers;
- *customer to customer (C2C)*, electronic transaction made between private persons, as occasional activity;
- *customer to business (C2B)*,regarding the relationship between individuals, registered as independent consultants or service supplier registered persons and companies, having a commercial aspect;
- *intra-business*, internal activities of a company or the relationship between different national affiliates and the corporation headquarters presented above;
- *government to citizen*, specific transaction used to reduce the costs, service improving and to assure the transparency.

A VE, is defined as “a temporary alliance between two or more independent partners associated in order to achieve a common goal”. The *virtual* term used in this definition *is not* referring to a non-existent organization and *is* referring to product image in end user eyes, which have in mind a single manufacturer for the product not an alliance. In today market, the efforts is concerning to *innovation capacity* (bringing new products or services on the market, changing the manufacturing strategy), *processing capacity* (new organization forms, integrated resource planning) and *cooperation capacity* (cross-organizational structure, network thinking, team working), in order to offer the needed product required in one time and one place.

The overall business requirements for e-economy environment is directly connected to the market challenges having as result the today’s enterprise need in order to satisfy a specific request appeared in market in one time and one place. To do so, specific goals must been defined:

- *changing the enterprise organization fashion*, adopting a matrix organization form, oriented to multi-polar project management, each business being seen as a project, having as result a product or a service;
- *limiting to a core expertise*, the global market impose a maximum efficiency for an enterprise, by out-sourcing all non-profit activities, keeping instead the most competitive activities in comparison with the concurrence;
- *concurrent process implementation*, segmenting and parallel process suite having as result reducing the time-to-market;
- *create multidisciplinary subsystems*, having multidisciplinary subsystems solve various problems that can occur in a project developing;
- *integration with partners*, develop an IT structure able to maintain the collaborative process with the partners involved in enterprise projects.

2 ICT impact in business

Offering opportunities to source a wide range of products from fragmented sources, e-Market places act as a potential catalyst for the internal market; they establish communities of buyers and sellers and mechanisms that allow enterprises to participate cost effectively in global markets. The most commonly category of application used in such great organization structure are (figure 1): CRM (Customer Relationship Management), ERP (Enterprise Resource Planning), PLM (Product Lifecycle Management) and SCM (Supply Chain Management).

For companies, thee-Economy has brought new opportunities, both within the IT industry and in other sectors, its impact varies substantially from sector to sector. e-Economy means that businesses can reach many more potential customers, work more effectively with other businesses and with governments. And they can use the new technology to change the way they work, modernizing their production processes and internal organisation, so that their operations can become more effective and efficient. Networked and virtual organisations, private and public, allow small entities to work together in a flexible way. For SMEs, the paradox is that "local" is becoming more not less important.

Accordingly with a recent study, the relevance of ICT in business is more then a single trend in all industry sector. The number of companies, which use the software application business is in continuous growth, giving a very good idea about the company’s competitiveness (figure 2).

3 Management structures

The IT expansion in business environment has changed the fashion of making business, covering on start “islands” of automation, now is view as an integrated system, being the backbone of the decision system. Considering this,we have to underline the requirements specific to e-economy. The integration of business processes has two directions: one among the organization, and second, the inter-organization axe. The “collaborative work” concept is regarding to business or business processes realized as participation of two or more organizations or organization entities (departments, services).The result of large IT implementation in business is an increasing of collaboration processes inside of a company or inter-company, converging to better economical results, by decreasing the Time to Market, using processes parallelism.

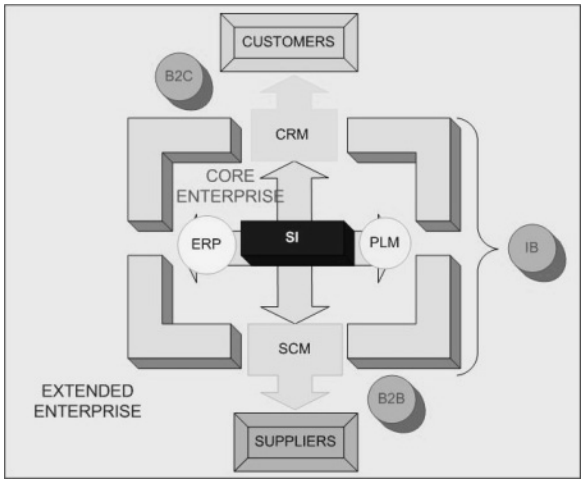


Figure 1: Relationship and software business chart in e-economy

First integration direction, inside organization, has appeared as response to increasing of problems complexity and decreasing of solving time.To solve more complex problem in a shorter time, require a multidisciplinary approach, with cross-organizational teams, having members from different activity fields. Thus, the adopted management structure is transiting from hierarchical to matrix form.

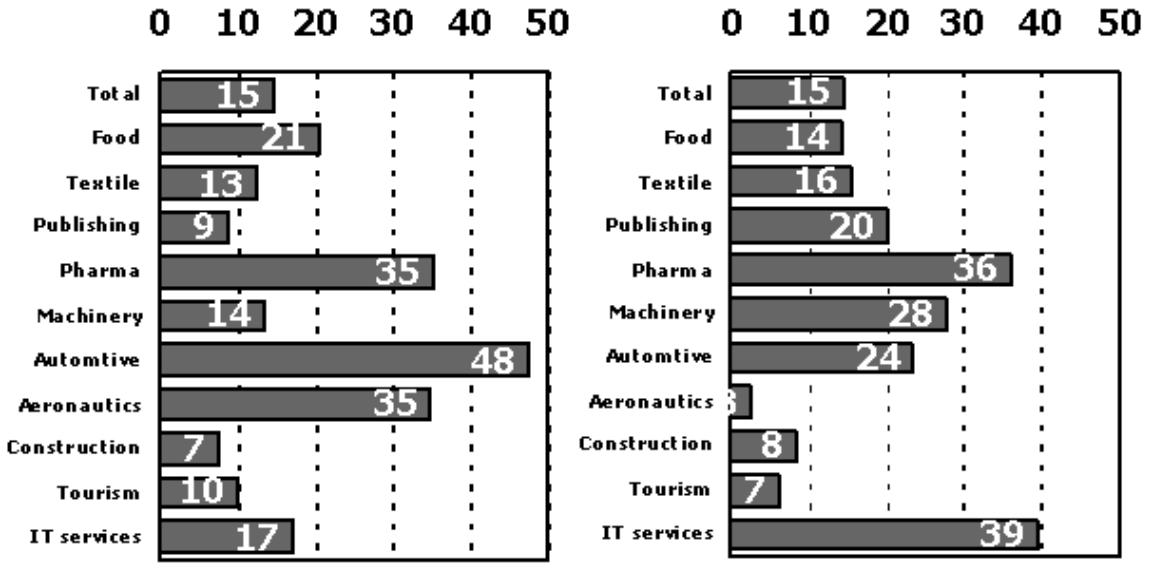


Figure 2: Companies using a SCM and CRM system in CZ, DE, ES, IT, PL, UK by sector [1]

In matrix organization form, each department is view as a sum of specialists, with equivalent competencies, forming the functional section (in department activity field) and specialists with no functional competencies (in department area of activity) used as an interface between department and other company's organizational structures (e.g. head of department, linking the department with top management). Also, in this structure, specific to collaborative process, are two subordination relationships, shown in figure 3 [2].

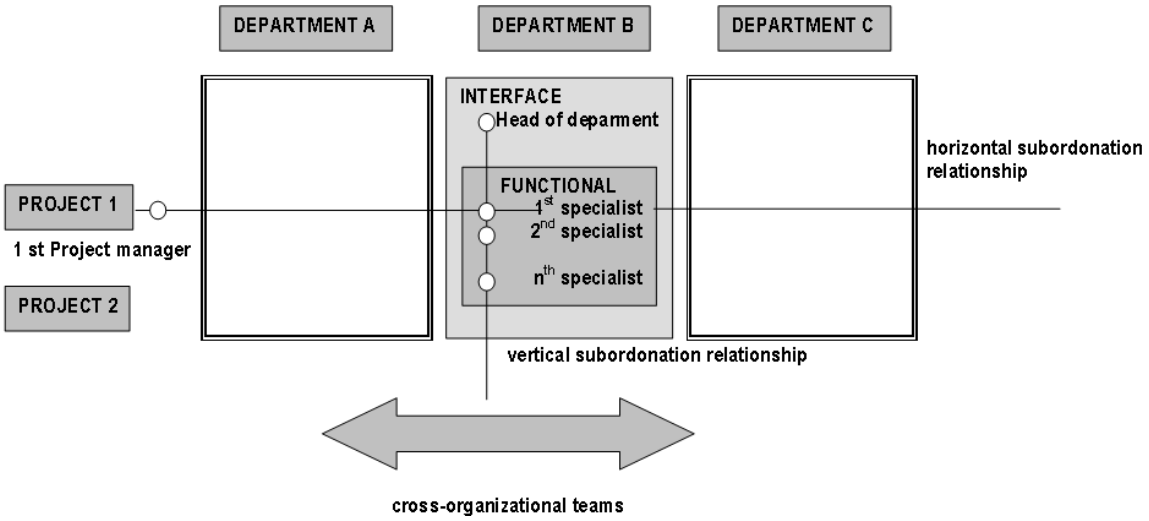


Figure 3: Matrix organizational management

4 European and Romanian e-business law frame integration

The e-economical activity has, basic, the same law frame as conventional economy, but, in addition, is completed with specific laws, because it complexity [3]. First important law was the Law No. 455/2001 regarding e-signature, which establishes general condition of e-signature implementation, for covering the contracts on distance specified on G.O. No. 130/2000. For g2c relationship, a big step forward was the adoption of Law No.291/2002 regarding the approval of taxes e-payment methods. Another important law is Law No.365/2002 regarding e-commerce activity developed in Romania.The Romanian legislation is completed after that with other important documents, such as Law No. 468/2002 regarding the approval of G.O. 20/2002having as subject the public electronic acquisition system, temporal stamp Law No. 451/2004 and Law No. 589/2004 regarding the juridical regime of e-legal representative notary activity. The above mentioned documents are in straight connection with European legislation, mentioning here the E-commerce Directive No.98/48 adopted by EC, completed by Distance Selling Directive, EC Framework Directive for Electronic Signatures, Electronic Money Directive and Data Protection Directive.

The EU recommend also the rules application establish by international entities as: WOT, UNCITRAL and OECD.Also, on European level, all e-commerce enterprises support the EU regulation,accordingly with EU Directive 2000/263. In conclusion, the European and,specially, Romanian law frame could maintain the e-activity on all it level,but, also true are the inconvenience related to difference exiting in each European member legislation and low level of application integration in Romanian enterprise (see Extended Enterprise model).

5 Summary and Conclusions

In current businesses, a large part of economical transactions are ’electronic’, and thus fall into the categories e-business or e-commerce. If at least one step in each phase is pursued electronically. e-business does not only describe external communication and transaction functions, but also relates to flows of information within the company, i.e., between departments, subsidiaries and branches. e-commerce refers to external transactions in goods and services. For consumers, thee-Economy brings many benefits. It means that products and services are available to people even in remote areas. And it means that consumers can compare what is offered by many different companies in many different places,to get the best deal. For employees, the e-Economy brings the prospect of more fulfilling knowledge-based jobs, and improvements in the way their working lives are organized Ű through different opportunities offered by e-Work.

This new business models leaded to the concept of Virtual Enterprises (VE) that is the foundation of the networked economy. The original goals for virtual enterprise business systems were to enable deployment of distributed business processes among different partners,to increase the efficiency of existing provided services, to

decrease the cost for the provision of these services, and to adapt rapidly to new market changes.

Dynamic virtual enterprises are rather a vision than nowadays reality. The vision encompasses small and highly skilled market players that are waiting for an opportunity to offer their services in their specific domain and work together with complementary partners to fulfill customer requirements.

The advantage of a flexible organization form lies in its capability to rapidly tackle problems and to react accordingly. It is not dependent on economies of scale but strives to provide be spoke and time-critical solutions according to the imposed requirements by every costumer and every endeavor.The prospect for dynamic VE is promising considering recent progresses of information and communication technology and the current trend to focus on a company's core competence. Hence, for a dynamic VE to succeed not only best-of-breed expertise in various domains is required, but also power of persuasion in respect of integrity and trustworthiness.

References

[1] http://www.ebusiness-watch.org/resources/charttool.htm, accessed January, 2006

[2] George Carutasu, Traian Aurite, "Parteners evaluation in virtual enterprise," *Scientific Bulletin of UPB,* D Series Vol. 77, pp. 25-36, 2005.

[3] Gomez Acebo, Pombo, "Information Society Law Approved" *International Law Office- Legal Newsletter*, http://www.internationallawoffice.com, September 2002.

George Căruţasu, Cornelia Botezatu
Romanian-American University
IT Management Systems Department
Address: B, Expozitiei Avenue, District 1, Bucharest
E-mail: georgecarutasu@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 118-123

Case Based Reasoning to Analyze Road Accidents

Valentina Ceausu, Sylvie Després

Abstract: In this paper is presented the prototype of a system designed to analyze road accidents. The analysis is carried out in order to recognize within accident reports general mechanisms of road accidents which represent prototypes of road accidents. The adopted problem solving paradigm is the case based reasoning. Natural language documents and semi-structured documents are used to create cases of our system. To cope with this difficulty we propose approaches integrating semantic resources. Hence, an ontology of accidentology and a terminology of road accidents are used to build descriptions of cases. The alignment of two resources supports the retrieval process. Based on models of accidentology, a data processing model is proposed to represent cases of the system. This paper presents the architecture of ACCTOS (ACCident TO Scenarios), a case based reasoning system prototype. The model to represent cases of the system is introduced and the phases of cases based reasoning cycle are detailed.

1 Introduction

In this paper the prototype of a system designed to analyze road accidents is presented. The analysis is carried out in order to recognize within accident reports general mechanisms of road accidents that represents prototypes of road accidents.

Case based reasoning is the adopted problem solving paradigm. Cased based reasoning solve a new problem by re-using a collection of already solved problem. The problem to solve is called target case. The collection of problems having already a solution represents the case base and it is a important feature of a case based reasoning system. The reasoning cycle of a case based reasoning system is composed of phases aiming to: create the target case; retrieve cases of the case base which are similar to the target case; adapt solutions of this cases, in order to propose a solution for the target case.

Natural language documents and semi-structured documents are used to create cases of our system. To cope with this difficulty approaches integrating semantic resources are proposed. An ontology of accidentology and a terminology of road accidents are used to build descriptions of cases. The alignment of two resources supports the retrieval process. Based on models of accidentology, a data-processing model is proposed to represent cases of the system.

The outline of this paper is as follows: first, the architecture of ACCTOS (ACCident TO Scenarios) is presented and the model proposed to represent cases of the system is introduced. Further, phases of cases based reasoning cycle are detailed. Conclusion and future work end this paper.

2 Architecture and resources of the system

To present the architecture, we use a division into modules, where each of the module addresses a different phase of the reasoning cycle (see Fig. 1).

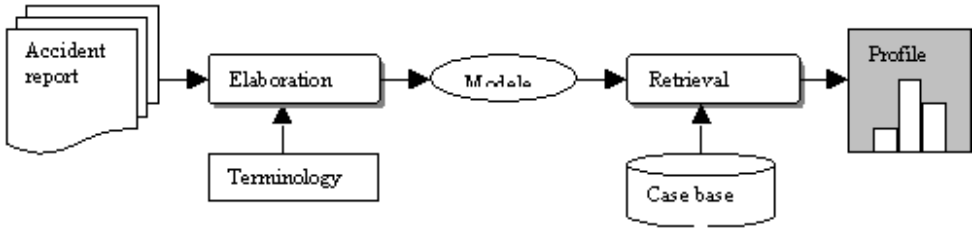


Figure 1: System architecture

Resources of the system

ACCTOS exploits two types of documents -accident reports and accident scenarios -to create cases.

Accident reports are documents created by the police. They include structured paragraphs describing the actors and the context of accident and natural language paragraphs explaining what happened in the accident (written with the help of witnesses, people involved in the accident or policemen).

Accident scenarios are documents created by researchers in road safety. They are prototypes of road accidents and present in a general way facts and causal relations between different phases leading to the collision. An accident scenario describes an accident as a sequence of 4 situations (or phases): the driven situation, the accident situation, the emergency situation and the shock situation. Prevention measures aiming to improve the road safety are provided for each accident scenario. A first study led by the department "Mechanisms of accidents" of INRETS (Institut national de recherche sur les Transport et leur Sécurité) established a first collection of accident scenarios involving pedestrians. Scenarios and assigned proposals will be used to build the initial case base.

Input of the system is a set of accident reports which have occurred on the same road section. Accidents are analyzed from the electronic form of accident reports. The PACTOL tool (Centre d'Etudes Techniques de l'Equipement de Rouen) made the reports anonymous. An electronic accident report is a semi-structured document containing structured paragraphs and natural language paragraphs. Structured paragraphs contains variables describing the accident.Variables correspond to humans and vehicles involved in the accident.The context of accident is also specified by variables. Text paragraphs describe what happened in the accident according to several points of view: police (synthesis of the facts), people involved (declarations) and witnesses.
From each accident report, a model is built by the "Elaboration" module. This model is used by the "Retrieval" module of the system in order to query the case base. The initial case based of ACCTOS is created from accident scenarios. As a result, correspondences between the initial accident report and accident scenarios are established. A correspondence is constituted of an assignment *accident report, accident scenario* and a trust assessment.

The output of the system is a profile of scenarios. A profile of scenarios is composed of several scenarios, where a coefficient in assigned to each scenario of the profile, reflecting its weighting within the profile.
The first module implements the authoring the case phase. Retrieval phase is done by the second module. Further on are presented the model proposed to represent cases of the system and phases of case based reasoning cycle .

3 Description of cases

Based on accidentology models (see [5]), a data-processing model is proposed to represent cases of the system. A case is described by two types of elements: global variables and agents.
Global variables specify the number of agents involved in accident, the environment in which the accident occurred - such as main road or secondary road - and context of the accident (by day, in intersection, etc.).
A human involved in accident and his vehicle represent an agent (see tab.1). This representation allows us to cope with difficulties related to metonymy between the human involved in accident and his vehicle. It also allows us to treat the particular case of pedestrian. Each agent is defined by his two components - human and vehicle - and by his evolution in accident. A domain term (ie: driver, car) and attributes (ie: age) are assigned to each component of an agent. Agent evolution is specified by a set of relations describing interactions: between components of an agent; between the agent and other agents involved in accident.

Agent	Humain	Vehicle	Attributes	Evolution
Agent 1	Pedestrian	no vehicle	age: 60	crossing; running
Agent 2	Driver	Car	age: 35	circulate; turning to

Table 1: Components of an agent

4 Authoring the case

The scope of this phase is to create the problem to solve, also called the target case.The model presented above is used to represented the target case. Each target case is created from an accident report. Both, the structured and the natural language paragraphs of an accident report are exploited to create the target case.

Identification of environnement

An accident report is a semi-structured document. Data about people and vehicles involved in accident and about the environment and context of the accident are stored in specific structures. Based on these structures we have created automatic procedures to retrieve valuable information.

Identification of agents

To describe an agent involved in accident we need to :

- Identify terms assigned to his components
- Identify values of his attributes
- Identify the evolution of the agent

Terms of components and values of attributes are identified automatically based on accident report structure. Agent evolution is identified based on natural language paragraphs of accident reports: declarations, testimonies, police synthesis. Agent evolution is expressed by a set of domain verbs identified within these paragraphs.
Text mining techniques and a terminology of road accidents are used jointly to identify the evolution of each agent. A terminology represents terms of a given field and relations between those terms. Relations are expressed by verbs and accepts, usually, two arguments: *Relation(domain, range)*, where *Relation* is a verb of the field, and *domain* and *range* are terms of the field.
For instance, *diriger-vers(véhicule, direction)* is a relation of the domain. We used a terminology created from 250 reports of accidents that occurred in and around Lille region. This terminology is expressed in OWL (see [15]).

Text mining techniques are also employed to identify evolution of agents. An approach based on information extraction using pre-defined patterns is adopted. We used lexical patterns to extract information. A lexical pattern is a set of lexical categories. For example *Noun, Noun* or *Verb, Preposition, Noun* are lexical patterns. In order to identify instances of patterns, natural language paragraphs are tagged using TreeTagger ([10]). A pattern recognition algorithm (see [2]) allows us to identify associations of words matching predefined patterns. The output of this algorithm is shown further:

Lexical Patterns and Corresponding word regroupings :
Noun, Preposition, Noun: groupe de piéton (group of pedestrians)
Noun, Preposition, Adjective: trottoir de droite (right side pavement)
Verb, Preposition, Noun: diriger vers place (direct to square)

We defined a set of verbal patterns which are able to highlight relations of the domain. A set R of verbal relations is extracted. Instances of those patterns could represents relations of the field, such as *diriger-vers (direct to)*, but also meaningless word regroupings, such as *diriger 306 (direct 306)*. They need to be validated and attached to agents of the accident. To do so, each agent $a(t_h, t_v)$, having t_h and t_v as components, query the terminology in order to identify relations having one of his components as argument. The result is:

$$R_{agent}(t_h, t_v) = R_{resource}(t_h) \bigcup R_{resource}(t_v)$$

, where $R_{resource}(t_h)$ and $R_{resource}(t_v)$ are relations of terminology having t_h, respectively t_v as arguments. By intersecting R_{agent} and R the evolution of an agent is identified as:

$$Evolution_{agent} = R_{agent} \bigcap R$$

Relations of R which are not modeled by the terminology are ignored. For each agent, the evolution is identified as set of verbal relations which are extracted from the accident report and validated by the terminology of road accidents.

5 Building the initial case base

The case base is a important feature of a case based reasoning system. Cases of the case base are called source cases and are represented by a couple $Problem, Solution$.
A set of accident scenarios is used to build the initial case base of the system. The accident scenario represents the "Problem"; measures of preventions assigned to the scenario represent the "Solution".
An **ontology of accidentogoly**(see [4]) supports description of source cases. This ontology was built from expert knowledge, texts of the field and accident scenarios. It models the concepts of the field and relations that hold between them. Concepts of ontology are structured in three main classes: concepts describing the human, the vehicle and the environment. A domain term and attributes are assigned to each concept. Concepts are connected by different types of relations. IS-A relations build the hierarchy of domain concepts. Verbal relations are also modeled that describe interactions between concepts.
An editor of scenarios was developed to build source cases. The editor integrates the ontology of accidentology. It allows to users to describe each accident scenario by choosing the appropriate concepts and relations of the ontology. The editor also allows users to assign to each concept or relation a coefficient indicating his importance. Importance coefficients are established based on linguistic markers. By integrating the ontology, homogeneous descriptions of cases are created.

6 Retrieval process

Retrieval process aims to retrieve source cases similar to the target case. Already solved problems similar to the target case are identified. By consequence, a solution can be proposed to the target case by adapting solutions of those problems. We propose a retrieval approach supported by the alignment of two semantic resources: the terminology and the ontology.
Ontology alignment can be described as follows: given two resources each describing a set of discrete entities (which can be concepts, relations, etc.), find correspondences that hold between these entities. In our case, a function $Sim(E_o, E_t)$ is used allowing us to estimate similarity between entities of ontology, E_o, and entities of the terminology, E_t, where an entity could be either a concept or a relation. Based on this, for T, a target case, two steps are needed to retrieve similar source cases.
(1) The first step is based on case base indexation. Global variables are used to index the case base. The values of global variables of the target case are taken into account to identify a set of source cases. The result is a set of source cases having the same context as the target case and involving the same number of agents.
(2)A voting process is used to improve this first selection. The vote is done by each agent of target case to express the degree of resemblance between himself and agents of a source case. A note is granted by each agent of target case to every source case. This note is granted by taking into accounts components of agents and theirs evolution. A first similarity measure proposed is given by:

$$Sim(a_i, a_j) = \begin{cases} SimComponent(a_i, a_j) + SimEvolution(a_i, a_j), & \text{if } SimComponent(a_i, a_j) \neq 0 \\ 0, & \text{if } SimComponent(a_i, a_j) \equiv 0 \text{'} \end{cases}$$

where a_i is an agent of the target case and a_j is an agent of a source case.
$SimComponent(a_i, a_j)$ express the similarity among the agents taking into account similarities of components:

$$SimComponent(a_i, a_j) = ch_j * sim(humain_i, humain_j) + cv_j * sim(vehicle_i, vehicle_j)$$

, where ch_j and cv_j are importance coefficients established for the source case, and values of $sim(humain_i, humain_j)$ and $sim(vehicle_i, vehicle_j)$ are given by the alignment of the two resources.

Evolution similarity express resemblances between evolutions of two agents:

$$SimEvolution(a_i, a_j) = \frac{\sum_r c_r * sim(rSource_r, rTarget_r)}{\sum_r c_r}$$

Coefficients c_r express the importance of $rSource_r$ relation for the considered source case. Values of $sim(rSource_r, rTarget_r)$ are given by alignment of the two resources.
Each agent of the target case evaluates his resemblance to agents of the source case by using the presented approach. A similarity vector is obtained. The note $note_i$ granted by the $agent_i$ to the source case is the maximum

value of this similarity vector. Based on notes granted by agents, the similarity between he target case and a source case is estimated by the average value:

$Sim(target, source) = \frac{\sum_{i=1}^{Na} note_i}{N_a}$
, where $note_i$ is the note granted by the agent $agent_i$, and N_a is the number of agents of the considered target case.

Indexing the case base allows a fast identification of source cases that are similar to the target case. By voting, the most similar cases are selected among the cases retrieved by the first selection. The retrieval process is driven by the description of source cases, whose importance coefficients are taken into account by similarity measures.

7 Summary and Conclusions

This paper presents the prototype of a system designed to analyze road accidents.Case based reasoning is the adopted problem solving paradigm. Cases of the system are created from semi-structured documents provided by two different communities: accident reports created by the police and accident scenarios created by researchers in road safety. Semantic resources are used to cope with heterogeneity and natural language representations. A terminology of road accidents supports authoring the case phase. Description of source cases is supported by the ontology of road accidents. Aligning the ontology of road accidents and the terminology supports the retrieval process.
This system is under development. Steps further are the implementation of proposed approaches, the system evaluation and possible improvements.
A few directions to improve the system can already be listed. The authoring the case phase could be improved by enriching the text mining techniques. Richer descriptions of target cases can be obtained.
The retrieval process id based on similarity measures taking into account common features of cases. Implementing dissimilarity measures, which will be based on differences between cases features could also be a further direction.

References

[1] R. Bergmann, "On the use of Taxonomies for Representing Case Features and Local Similarity Measures" *Proceedings of the 6th German Workshop on Case-Based Reasoning* , pp. 22-31, 1998.

[2] V. Ceausu, S.Desprès, " Towards a Text Mining Driven Approach for Terminology Construction" *Proceedings of the 7th International conference on Terminology and Knowledge Engineering*, pp. 63-72, 2005.

[3] H. Cherfi, A. Napoli, Y. Toussaint, " Vers une méthodologie de fouille de textes s'appuyant sur l'extraction de motifs fréquents et de règles d'association" *Proceedings of Conférence francophone sur l'apprentissage automatique*, pp. 68-70, 2003.

[4] S. Desprès, *Contribution à la conception de méthodes et d'outils pour la gestion des connaissances* Université René Descartes, 2002.

[5] D. Fleury, *Sécurité et urbanisme. La prise en compte de la sécurité routière dans l'aménagement urbain*, Presses de l'Ecole Nationale des Ponts et chaussées, 1998.

[6] K.M. Gupta, D.W. Aha, N. Sandhu," Exploiting taxonomic and causal relations in conversational case retrieval" *Proceedings of the Sixth European Conference on Case-Based Reasoning*, pp. 133-147,2002.

[7] M. Klein," Combining and relating ontologies: an analysis of problems solutions" *Workshop on ontologies and Information sharing, IJCAI '01*, pp. 309-327,2001.

[8] U. Hahn, K. Schnattinger, "Towards text knowledge engineering" *Proceedings of AAAI'*, pp. 129-144, 1998.

[9] P. Seguela," Adaptation semi-automatique d'une base de marqueurs de relations sémantiques sur des corpus spécialisés " *Actes de Terminologie et Intelligence Artificielle*, pp. 52-60, 1999.

[10] H. Schmid, " Probabilistic part-of-speech tagging using decision trees " *In International Conference on New Methods in Language Processing*, pp. 73-82, 1994.

[11] B. Smyth, P. McClave, " Similarity vs. diversity" *Proceedings of the 4th International Conference on CBR*, pp. 347-361, 1994.

[12] R. Weber, D.W. Aha, N. Sandhu, H. Munoz-Avila , " A textual case-based reasoning framework for knowledge management applications " *In Proceedings of the Ninth German Workshop on Case-Based Reasoning*, pp. 84-96, 2001.

[13] N. Wiratunga, S. Craw, S. Massie, " Index Driven Selective Sampling for CBR " *In proceedings of the 5th International Conference on Case-Based Reasoning*, pp. 71-79, 2003.

[14] N. Wiratunga, I. Koychev, S. Massie, "Feature Selection and Generalisation for Retrieval of Textual Cases" *In Proceeding of the 7-th European Conference on Case-Based Reasoning*, pp. 73-82, 2004.

[15] World Wide Web Consortium (W3C), "OWL - Web Ontology Language", http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

Valentina Ceausu, Sylvie Després
René Descartes University
Mathematics and Computer Science Departement
Address: 45 rue des Saints Pères
75270 Paris cedex 06
E-mail: {valentina.ceausu,sd}@math-info.univ-paris5.fr

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 124-129

A Multi-Agent System for Design Information Management and Support

Camelia Chira, Ovidiu Chira

Abstract: Enterprise models of engineering design involve multiple distributed design teams with heterogeneous skills cooperating together in order to achieve global optima in design. The success of this distributed design organization depends on critical factors such as the efficient management of the design related information circulated in the distributed environment and the support for the necessary cooperation processes among participants dispersed across the enterprise. This paper proposes a multi-agent design information management system to aid the new enterprise model by supporting the synthesis and presentation of information to distributed teams for the purposes of enhancing design, learning, creativity, communication and productivity. Enabled by multi-agent systems and information ontologies, the proposed system facilitates interoperation among distributed resources and knowledge sharing, reuse and integration in a distributed design environment.

Keywords: multi-agent systems, distributed engineering design, knowledge management

1 Introduction

Emerging in response to market demands and competitive pressures, the distributed engineering design organization requires multidisciplinary design teams to virtually collaborate in order to achieve the global optimal design solution [1], [2]. The human and information resources involved in the design process are geographically, temporally, functionally and/or semantically distributed in a virtual environment [3]. This new enterprise model has the potential of achieving benefits such as the generation of new insights, new ideas and improved design solutions enriched by the multiple skills of the designers engaged in the design task and easier access to multiple sources of information. Because the communication of information, the coordination of engineering design participants and team collaboration takes place in a computer based medium, the availability of the software infrastructure to support cooperation and facilitate the management of various design information structures remains the key success factor of distributed design.

A Multi-Agent Design Information Management and Support System (MADIS) is proposed to aid the distributed engineering design organization by efficiently enabling the management of the data-information-knowledge value chain. The proposed system employs multi-agent systems to enable interoperation among distributed resources and information ontologies for knowledge sharing, reuse and integration. Consisting of a collection of autonomous software agents able to cooperate with each other, the proposed system supports the designer's decision-making process in a distributed environment by facilitating the storage, retrieval, exchange and presentation of data, information and knowledge. Furthermore, cooperation among multidisciplinary design teams is aided by flexible graphical user interfaces, a common shared knowledge base and easy access to relevant and timely information.

2 Multi-agent systems

Software agents and Multi-Agent Systems (MAS) represent an important and fast growing area of AI and more generally of Computer Science [4], [5]. Capable of managing the complexity inherent in software systems, MAS represent the appropriate solution for domains in which data, control, expertise and/or resources are inherently distributed [6].

The agents in a MAS system must *coordinate* their activities (to determine the organisational structure in a group of agents and to allocate tasks and resources), *negotiate* if a conflict occurs and be able to *communicate* with other agents. Furthermore, because agents may have different terms for the same concept or identical terms for different concepts [7], a meaningful communication process among agents requires, besides an Agent Communication Language, a common understanding of all the concepts exchanged by agents. *Ontologies* [8], [9] represent the technology to support this requirement by semantically managing the knowledge from various application domains.

Many research studies emphasize the need to use ontologies for domain knowledge representation in order to meaningfully support agent interoperation [8], [9]. The study of ontologies has developed gradually from specific needs associated with the problem of knowledge management within a computational environment and particularly from the problem of knowledge sharing and reuse (emerged within AI) [10]. Ontologies overcome the difficulties raised by "monolithic, isolated knowledge systems" [11], by specifying content specific agreements to facilitate knowledge sharing and reuse among systems that submit to the same ontology/ontologies by the means of ontological commitments [12].

3 A multi-agent design information management architecture

The overall objective of the proposed MADIS system is to support the distributed design process by managing information, integrating resources dispersed over a computer network and aiding collaboration processes. It is intended to design a multi-agent system composed of several interacting agent sub-systems in order to deliver these goals. Furthermore, the information circulated within the distributed design environment will be stored in an ontology library to enable content-related support for information management. The multi-agent approach to distributed engineering design coupled with the use of ontologies promises to tackle important distributed design issues such as interdisciplinary cooperation among distributed designers, exchange of design data, information and knowledge and integration of heterogeneous software tools [10], [15].

The MADIS system employs agents for information storage and retrieval, for enhancing collaboration within a distributed design environment and for providing a suitable interface to the user. The efficient performance of these tasks is ensured by the cooperation process among the different kinds of agents that form the system. These agents can be divided in four societies i.e., User management, Application management, Ontology management and Agent interconnection and management (see Figure 1). Supporting agent interoperation, the FIPA[1] agent management ontology is part of each agent expertise. The agent communication language used is FIPA ACL, based on which agents are able to exchange messages (of types such as request, query and inform) in order to achieve different objectives. Furthermore, the MADIS ontology completes the expertise of the basic MADIS agents.

The agents from the *user society* form the interface between the system and the designer. They provide different services to the user and respond to queries and events initiated by the user (or on behalf of the user) with the help of the ontological agents. Each designer is served by a *User Profile Manager* agent, which stores and manages the user description and preferences based on which various services are offered. This agent should learn from the user as to improve the functionality of the system over time. Furthermore, a *User Interface Controller* agent directly assists the user in his/her tasks through a graphical interface. This agent enables user access to different services mainly based on the cooperation with the User Profile Manager agent and the ontology agent society.

The agents from the *application society* have the capability to retrieve information from the applications used by the designer and forward it for storage to the ontological agents. The information circulated in the computer-defined distributed design environment is usually available in the various proprietary applications used by the designers to support them in their tasks. In order to make this heterogeneous information readily available to distributed users and to semantically integrate dispersed resources, each such application is controlled through an Application Controller agent. This agent has to be first integrated in the application served and then forward all the information that can be extracted using the API to the ontology agent society for storage purposes. The Application Controller agents can act autonomously to achieve their objectives or can be controlled by the user (through the User Profile Manager) who can set different functional parameters if desired. Depending on the flexibility of the specific API, each Application Controller agent can extract information about a number of different conceptual structures from the current application (e.g., assembly structural information from a Catia system) and transform it into an internal format.

The agents from the *ontology society* provide ontology management services. They are able to access, retrieve, add, modify and delete information from the ontology library. Creating semantic link among the system's architectural components, the MADIS ontology describes concepts, relationships and inference rules of the engineering design domain. The scope of the MADIS engineering design ontology is to create a common shared understanding of the application domain so that information and knowledge can be shared among the members of the distributed design environment.

The ontology agent society contains different kinds of agents able to maintain (e.g., add, delete, modify) the information structures stored in the MADIS ontology. The *Ontology Manager* agent supervises the ontology

[1] Foundation of Physical Intelligent Agents – http://www.fipa.org

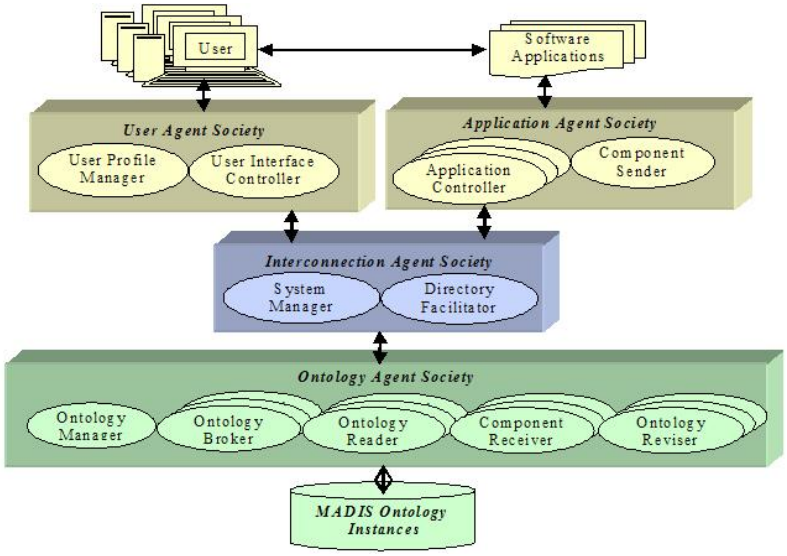


Figure 1: The MADIS agent societies

management process ensuring both the accuracy and consistency of the ontology and that requested ontology-related services are delivered. The *Ontology Broker* agent manages the agents that can read the ontology (i.e., the Ontology Reader agents) and the services provided by them. The *Ontology Reader* agents are mobile agents able to read the ontology and arrange the mined information in graphical format. After the user interface containing the requested information has been created, the Ontology Reader migrates to the initial requester agent (e.g., a User Interface Controller agent). The *Component Receiver* agents have the capability of adding new instances of specific concepts (or components) to the corresponding ontology from the MADIS ontology library.

The *interconnection society* contains agents that supervise and support the interoperation process among the other MADIS agents. The main objective of this agent society is to ensure that MADIS agents are meaningfully interconnected. This is achieved by a *System Manager* agent that supervises the overall functionality of the multi-agent system and a *Directory Facilitator* agent that helps MADIS agents to find the agent(s) that provides a requested service.

The agent interactions within MADIS are vital for a successful and constructive support provided to the distributed designer. As already indicated, MADIS agents are FIPA compliant and communicate by exchanging ACL messages. To exemplify MADIS agent interaction, Figure 2 presents the AUML protocol diagram [13] of the *User-Request-Information* scenario, which occurs each time the user wants to browse or to search the MADIS ontological instance base.

The User-Request-Information scenario involves the following main steps i.e., (1) the User Interface Controller queries the User Profile Manager for the services provided to the user through a FIPA-QUERY protocol, (2) the User Interface Controller queries the Ontology Manager for the concept categories available in the ontology that can be accessed by the user, (3) the User Interface Controller requests the Directory Facilitator for the identification of the agent that can provide the service requested by the user, (4) the User Interface Controller requests the Ontology Broker (identified in the previous step) for the service (e.g., browse, search) needed by the user, and (5) the Ontology Broker instantiates the appropriate Ontology Reader mobile agent that will fulfil the requested service and will migrate back to the User Interface Controller location with the result.

4 Implementation and testing

The aim of the implementation phase is to provide a working prototype model of MADIS that can exemplify and demonstrate the purpose and validity of the system and that can be analysed and evaluated in the testing and validation phase [15]. The programming language selected for implementation is Java due to its rich library of functions tackling concurrency, code portability, native support for multithreading and introspection of object properties and methods. Furthermore, the Java Agent DEvelopment Framework (JADE) enables the implementation of agent interoperation within MADIS. The current MADIS prototype contains Application Controller agents

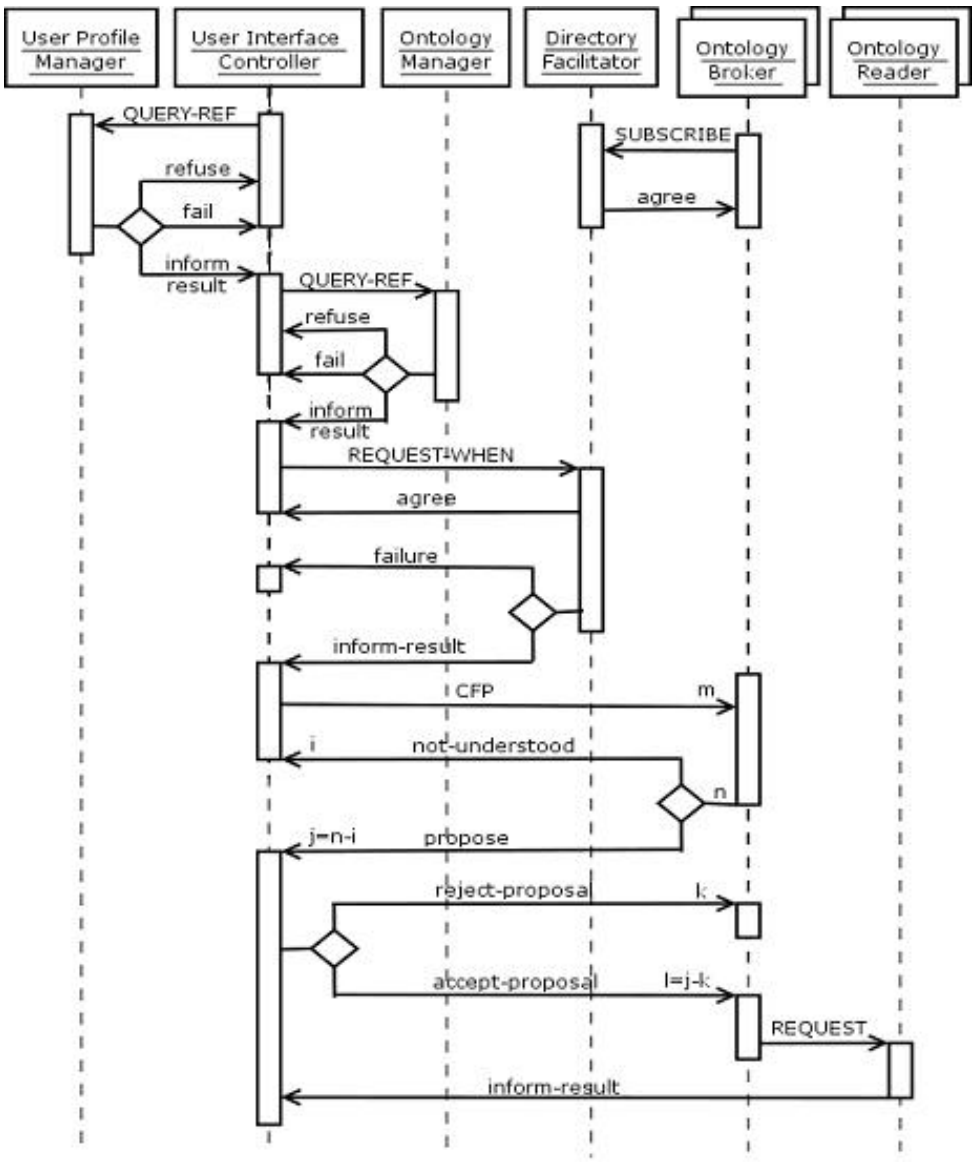


Figure 2: The User-Request-Information AUML interaction protocol diagram

integrated in a CAD tool called ProEngineer 2001. The MADIS ontology was implemented using the Resource Description Framework (RDF) and RDF Schema (RDFS) infrastructure with the support of the Protege editor tool.

The testing phase of MADIS used the protocol analysis (PA) technique [14] to evaluate the proposed system when used by a single designer or by a team of designers in a distributed environment to perform a given set of tasks. The subjects were videotaped while using the system and verbalizing their thoughts or communicating with other designers (depending on the task). The PA test results show that agent properties such as autonomy, pro-activeness, cooperation and mobility are highly beneficial to the distributed designer during the information-intensive problem solving process of design. Compared to traditional groupware technologies (e.g., Sametime Document Repository), the multi-agent approach has clear potential benefits including reliability, robustness, faster access to required information.

5 Conclusions and future work

Including the phases of system specification, design, implementation and testing, the development of the proposed Multi-Agent Design Information Management and Support System (MADIS) addresses the key information needs of distributed collaborative engineering design. The MADIS multi-agent system employs cooperating agents that can support the user through learning, autonomous agents for information retrieval, mobile agents to address various designer needs, web interfaces for easy access to design knowledge and ontologies for semantic management of design information structures. Offering computational efficiency, dependability and flexibility, multi-agent systems coupled with ontologies represent a promising approach to support the design process in a distributed collaborative design environment facilitating interoperation among distributed resources, interdisciplinary cooperation and information sharing.

Further research into the areas of human-computer interaction, human designer and semantic web can potentially deliver many benefits and major improvements for the application of AI technologies to distributed engineering design. Future work should also focus on the extension of MADIS to an intelligent system that supports and improves the distributed engineering design process and also has the capability to trigger designers creativity and encourage new ideas and perspectives.

References

[1] B. Hirsch, "'Extended Products in Dynamic Enterprises"', *E-Business: Key Issues, Applications and Technologies*, pp. 622-628, 2000.

[2] S.P. MacGregor, "'New Perspectives for Distributed Design Support"', *The Journal of Design Research*, Vol. 2(2), 2002.

[3] F. Pena-Mora, K. Hussein, S. Vadhavkar, K. Benjamin, "'Cairo: A Concurrent Engineering Meeting Environment for Virtual Design Teams"', *Artificial Intelligence in Engineering*, 14: 202-219, 2000.

[4] H.S. Nwana, "'Software Agents: An Overview"', *Knowledge Engineering Review*, Vol. 11(3), pp. 1-40, 1996.

[5] J.M. Bradshow, "'An Introduction to Software Agents"', in *Software Agents*, Bradshow JM Editor, MIT Press: Cambridge, 1997.

[6] N.R. Jennings, "'On Agent-Based Software Engineering"', *Artificial Intelligence*, 2000.

[7] J. Odell, "'Agent Technology - Green Paper"', *OMG - Agent Platform Special Interest Group*, 2000.

[8] P. Borst, H. Akkermans, J. Top, "'Engineering Ontologies"', *International Journal of Human-Computer Studies*, Vol. 46 (Special Issue on Using Explicit Ontologies in KBS Development), pp. 365-406, 1997.

[9] R. Studer, V.R. Benjamins, D. Fensel, "'Knowledge Engineering: Principles and Methods"', *Data and Knowledge Engineering*, Vol. 25(1-2), pp. 161-197, 1998.

[10] C. Chira, D. Tormey, T. Roche, A. Brennan, "'An Ontological and Agent Based Approach to Knowledge Management within a Distributed Design Environment"', *First International Conference on Design Computing and Cognition*, MIT, Cambridge, USA, 2004.

[11] T.R. Gruber, "'The Role of Common Ontology in Achieving Shareable, Reusable Knowledge Bases. Principles of Knowledge Representation and Reasoning"', *Proceedings of the Second International Conference*, San Mateo, Morgan Kaufmann, 1991.

[12] P. Spyns, R. Meersman, M. Jarrar, "'Data Modelling Versus Ontology Engineering"', *ACM SIGMOD Record*, 2002.

[13] B. Bauer, J.P. Müller, J. Odell, "'Agent UML: A Formalism for Specifying Multiagent Interaction"', *Agent-Oriented Software Engineering*, Springer-Verlag, Berlin; 2001.

[14] K.A. Ericsson, H.A. Simon, *Protocol Analysis: Verbal Reports as Data*,1999.

[15] C. Chira, *The Development of a Multi-Agent Design Information Management and Support System*, PhD Thesis, Galway-Mayo Institute of Technology, 2005.

Camelia Chira
Babes-Bolyai University, Cluj-Napoca
Department of Computer Science
Address: Str. M. Kogalniceanu, No 1B, 400084, Cluj-Napoca, Romania
E-mail: cchira@cs.ubbcluj.ro
Ovidiu Chira
Galway-Mayo Institute of Technology, Galway
Department of Industrial Engineering
Address: Str. Dublin Road, Galway, Ireland
E-mail: ovidiu.chira@gmit.ie

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 130-134

Performance Study of Receiver Diversity Techniques in 802.11a WLANs

Ligia Chira, Tudor Palade

Abstract: Our analysis is performed in the context of a study of adaptive radio techniques, in the attempt to improve link availability and transmission quality of broadband wireless systems, and especially to achieve an efficient use of radio resources. We have established our focus on spatial diversity techniques, and this paper presents a performance analysis of receiver combining techniques for indoor multipath scenarios, in the particular case of an 802.11a PHY. Even if a performance hierarchy of these techniques is established - based on diversity gain improvement - we wanted to see which combining technique is better for a particular scenario. The detailed observations will enable us to set the premises for new adaptive techniques or adaptive combinations.

Keywords: space diversity, receiver combining, multipath propagation, 802.11a PHY.

1 Introduction

The context of our analysis is a study of adaptive techniques used in current wireless systems. The extreme variability of wireless channels demands that adaptivity be the one constant feature of high-performance systems [1].

The variability of wireless channels presents both challenges and opportunities in designing multiple access communication systems. To maximize throughput for a given power budget, the link must adapt to the actual channel conditions, changing the transmitter power level, antenna beam pattern, equalizer settings, and possibly the symbol rate and constellation size [1].

We have established our focus on physical and link layer adaptation, and we have identified the adaptive potential of receiver combining techniques. Providing services in NLOS (Non Line-of-Sight) conditions relies on using the multipath signals to an advantage. The 802.11a simulation platform [2] that we have used in our studies, already benefits from the advantages of OFDM (Orthogonal Frequency Division Multiplexing)- high spectral efficiency, resilience to RF interference, and lower multipath distortion. The ability to efficiently overcome delay spread, multipath, and ISI allows for higher data rate throughput.

Spatial diversity techniques are efficient in fighting multipath propagation effects and we will show how this can be achieved in the particular case of the 802.11a PHY. Diversity gains evaluated for different data rates are reported in [3], for eight antennas. Switch diversity is reported to provide a gain of up to 2 dB - corresponding to a 20% range improvement. For Selection Diversity, the gain lies between 7-9 dB, meaning 50% system range improvement. Maximal Ratio Combining provides an even higher diversity gain, 12-16 dB depending on the chosen data rate, and the system range is improved with up to 100%.

2 Receiver Combining Techniques

There are several types of receiver diversity combiners, with different implementation complexity and overall performance. We have tested four of them: Threshold Combining (ThC), Selection Diversity Combining (SDC), Maximal Ratio Combining (MRC), and Equal Gain Combining (EGC), and this paper presents the analysis of the first three, with a focus on ThC.

In Threshold Combining (ThC) the received signals are scanned in a sequential order, and the first signal with a SNR level above a certain threshold is selected. This signal is used as long as its SNR is higher than the threshold value. When it falls below the threshold the selection process is reinitiated. With only two-branch diversity this is equivalent to switching to the other branch when the SNR on the active branch falls below SNRth. This method is called switch and stay combining (SSC) [4]. Since the SSC does not select the branch with the highest SNR, its performance is between that of no diversity and ideal Selection Combining [4].

In Selection Diversity Combining (SDC), the SNRs of the received signals are continuously monitored so that the output of the combiner has a SNR equal to the maximum SNR of all the branches. SDC does not require co-phasing of multiple branches since only one branch output is used. To work properly each antenna branch

must have relatively independent channel fading characteristics. To achieve this, the antennas are either spatially separated, use different polarization, or a combination of both.

In Maximal Ratio Combining (MRC) each signal is given a gain proportional to the ratio between the fading amplitude and the noise power. This technique proposes a means of combining the signals from all receiver branches, so that signals with a higher received power have a larger influence on the final output. Since the signals are summed they must have the same phase to maximize performance. This requires not only separate receivers but also a co-phasing and summing device.

Equal Gain Combining (EGC) is a type of MRC. The weighting is equal, the weights are all set to the same value and are not changed after that. Then the signals are co-phased before the summation process just like in MRC. Equal gain combining is used for non-coherent systems when it is difficult to obtain accurate channel estimates (e.g., a fast frequency-hopped system). Its performance is not much better than selection diversity, but it may be used even for quickly changing channels. A separate receiver is required for each antenna, since the combining takes place after demodulation [1].

3 Simulation-based analysis

Our study is focused on indoor multipath propagation scenarios, in dispersive fading conditions. Simulation settings include 3 to 12-path scenarios.

The channel block of the 802.11a PHY platform, which is a multipath Rayleigh channel, allows the setting of delays and gains for each path. According to the standard [5] we have chosen the indoor specification with delays between 40ns and 200ns.

The platform employs a link adaptation scheme wherein we select the best coding rate and modulation scheme based on channel conditions. The Adaptive Modulation Control block adapts the transmission to channel variations by switching between the available modulation schemes: BPSK (1/2, 3/4), QPSK (1/2, 3/4), 16QAM (1/2, 3/4), 64QAM (2/3, 3/4) according to the 802.11a PHY specifications. As the channel quality becomes poorer, the type of modulation changes to a more robust one. All mandatory and optional data rates are available: 6, 9, 12, 18, 24, 36, 48, and 54 Mbps.

Other simulation settings are: the symbol period, 0.08us, the number of OFDM symbols per block, 20 symbols, the number of OFDM symbols in the training sequence, 4 symbols, the hysteresis factor for adaptive modulation, 3 dB, the Viterbi traceback depth, 34, and the maximum Doppler shift, 50 Hz.

Figure 1 synthesizes the simulation results for reference scenarios, when no combining is used, for up to 12 propagation paths. As the number of propagation paths is increased the overall performance degrades. We can notice that the main problem are the high PER (Packet Error Rate) values obtained when simulating a higher number of paths and especially the fact that the non-zero PER values tend to last longer. Operation limiting values for these parameters have been established in [6].

Number of propagation paths	**Mean SNR [dB]**	**Mean Bit Rate [Mbps]**	**Mean PER [%]**
2	24.83	37 [24-54]	4
3	23.7	34 [24-36]	7.82
4	23.7	33.5 [24-36]	8.2
6	23.1	32 [24-36]	5.72
8	22.84	30.7 [24-36]	6.48
10	22.6	30.35 [24-36]	6.81
12	22.84	31.33 [24-36]	7.34

Figure 1: Simulation results for reference scenarios (2 to 12 propagation paths)

Figure 2 presents a comparison of reference simulation values and values obtained for three combining techniques: ThC, SDC, and MRC. The notations ThC2, ThC3, ThC4, SDC2, SDC3, MRC3 contain the name of the technique and the number of receive antennas. SNRth is the SNR threshold value which can be set for ThC, and which greatly influences the performance of the combining technique. The right brackets indicate that the bit rate values are concentrated in that interval.

Receiver Combining Technique	Mean SNR [dB]	Mean Bit Rate [Mbps]	Mean PER [%]
None $_{2\ paths}$	24.83	37 [24-54]	4
None $_{3\ paths}$	23.7	34 [24-36]	7.82
None $_{4\ paths}$	23.7	33.5 [24-36]	8.2
ThC2 $_{SNRth=10dB}$	27.4	43.8	0.9
ThC3 $_{SNRth=10dB}$	27.7	44.5	0.66
ThC4 $_{SNRth=10dB}$	27.5	43.86 [36-48]	0.9
SDC2	27.75	44.5	0.12
SDC3	24	35	0.27
MRC3	21.75	29 [24-36]	0.77

Figure 2: Simulation results for three combining techniques

Looking at the fourth column in figure 2 we can notice that receiver combining techniques improve performance in terms of PER, thus ensuring a higher link availability and reliability. Figure 3a clearly illustrates the improvement in PER when using one of the receiver combining techniques.

SDC and MRC are the schemes that ensure the lowest PER. When using ThC all the monitored parameters are influenced by the selected SNR threshold value and we will discuss this in the next paragraphs. SDC has the advantage that it selects the actual best signal in terms of received power, and not merely the first one that meets a certain criteria.

SDC has the advantage of ensuring a low PER and the disadvantage of moderate bit rates (fig.3b), compared to those provided by ThC, because this technique performs the calculations on the low SNR branches too, and does not cancel them like ThC does. Still SDC outperforms ThC in terms of overall diversity gain and coverage.

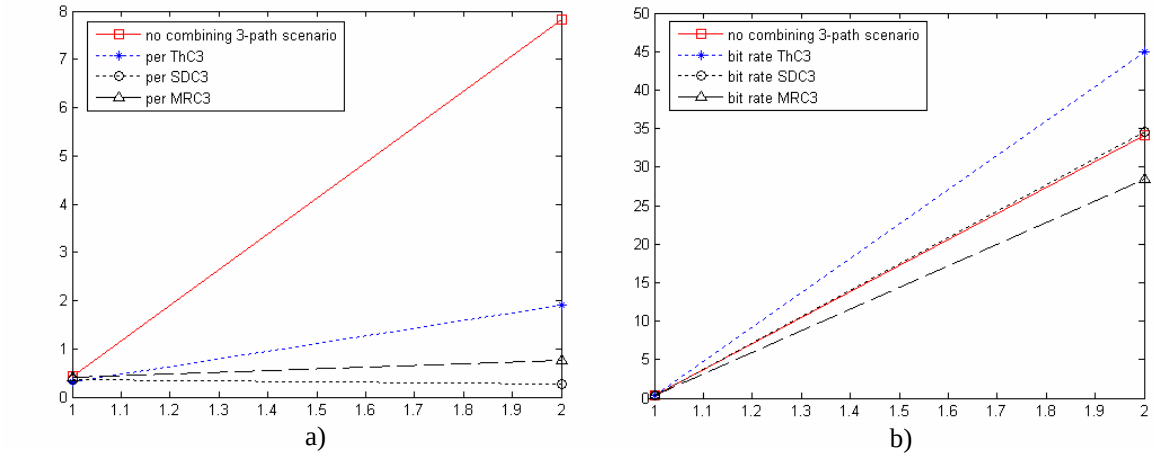


Figure 3: a) Mean PER using combining techniques for a 3-path scenario; b) Mean bit rate using combining techniques for a 3-path scenario

We can notice that the highest bit rates are obtained when using ThC (fig.3b), as this technique cancels the poor SNR branches according to its threshold. Threshold combining enables high bit rates (in the range of 36-54Mbps), and brings significant improvement in terms of PER also. Choosing the threshold value is a compromise between bit rate and PER values. A major drawback of ThC is that some other branches may have an even better SNR than the chosen branch, and still they can be suppressed. To mitigate this disadvantage, an optimised threshold has to be found.

We found it very interesting to analyse the influence of the number of receive antennas and of the threshold value on ThC performance. Figure 4 contains the simulation results for several scenarios in this respect.

We can notice that a higher threshold (e.g. 15-20dB) ensures high bit rates but we still get rather high PER values - a mean of 2-7%, because weaker branches are ignored even if they could be useful in that particular difficult situation. What may seem an advantage - the fact that a very low SNR branch does not influence the output of the receiver, as only one branch at a time is selected - could turn into a disadvantage when the link quality gets very poor and there is no better branch to choose. A lower threshold seems to get both advantages:

Receiver Combining technique		Mean SNR [dB]	Mean Bit Rate [Mbps]	Mean PER [%]
None	2 paths	24.83	37 [24-54]	4
	3 paths	23.7	34 [24-36]	7.82
	4 paths	23.7	33.5 [24-36]	8.2
ThC2	SNRth=10dB	27.4	43.8	0.9
ThC3	SNR_{th}=25dB	28.7	50.7 [48-54]	24.3
	SNR_{th}=20dB	28	46 [36-54]	5.88
	SNR_{th}=15dB	28	45	1.9
	SNR_{th}=10dB	27.7	44.5	0.66
	SNR_{th}=5dB	27.58	44.2	0.26
ThC4	SNR_{th}=20dB	28	46.5 [36-54]	6.83
	SNR_{th}=15dB	27.5	44 [36-48]	2.27
	SNR_{th}=10dB	27.47	43.86 [36-48]	0.9
	SNR_{th}=5dB	27.5	44	0.19

Figure 4: ThC performance function of the number of receive antennas and value of the SNR threshold

high bit rates - a mean of 44-45Mbps - and low PERs - a mean of 0.19-2.27% - meaning less frequent and less high PER rise.

Figure 5a illustrates the influence of the chosen threshold on the achieved bit rate.

For the 3-path scenario it can be noticed that a high SNR threshold (e.g. 25dB) ensures higher bit rates, all in the range of 48-54 Mbps (fig 4). The problem is that the PER values obtained when applying this threshold are unacceptable - a mean of 24.3%, (fig 4, fig. 5b). This is only normal because many potentially useful signals are cancelled just because their SNR is under this high threshold. A low SNR threshold (e.g. 5dB) still enables high bit rates - a mean of 44.2 Mbps, and at the same time very good PERs -a mean of 0.26%, and a maximum of 16%.

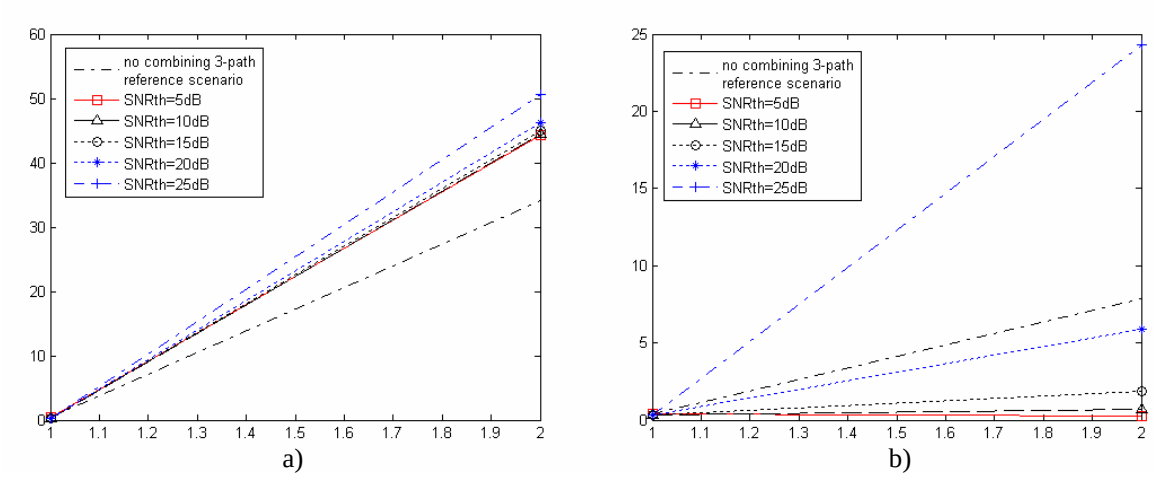


Figure 5: a) Mean bit rate function of SNR threshold value for 3-path Threshold Combining; b) Mean PER function of SNR threshold value for 3-path Threshold Combining

4 Conclusions

Our simulations show that receiver combining techniques improve performance in terms of PER, thus ensuring a higher link availability and reliability. The performance of the diversity combiners increases with the number of antennas, but not linearly, and will eventually stop growing beyond a certain number of antennas. The highest gain is obtained by passing from a non-diversity scheme to a two-branch spatial diversity scheme. Increasing the number of branches from two to three will yield a much lower gain then by switching from one to two branches and in general increasing the number of receivers yields less significant SNR gain improvement.

SDC and MRC are the schemes that ensure the lowest PERs. When using ThC all the monitored parameters are influenced by the selected SNR threshold value.

The MRC technique is more efficient in the case of non-dominant-path scenarios where the SNR values of the various propagation paths are similar. Also, this technique is more suited for uplink implementation in the base station receiver.

SDC yields better results for dominant-path scenarios and has the advantage of ensuring a low PER and the disadvantage of moderate bit rates, compared to those provided by ThC, because this technique performs the calculations on low SNR branches too, and does not cancel them like ThC does. Still SDC outperforms ThC in terms of overall diversity gain and coverage.

In case we use the ThC technique for a dominant-path scenario the SNR threshold value should be chosen very carefully, otherwise we risk to cancel the dominant path. Threshold combining enables high bit rates (in the range of 36-54Mbps), and brings significant improvement in terms of PER. Choosing the threshold value is a compromise between bit rate and PER values. A major drawback of ThC is that some other branches may have an even better SNR than the chosen branch, and still they can be suppressed. To mitigate this disadvantage, an optimised threshold has to be found. We have analysed the influence of the number of receive antennas on ThC performance and we have obtained the lowest PER values, and still high bit rates, for ThC4. We have also analysed the way the SNR threshold influences the bit rate and PER performance of the ThC technique, and we can conclude that a lower threshold enables both high bit rates and low PERs.

References

[1] Gregory J. Pottie, ”Wireless Multiple Access Adaptive Communications Techniques”, Electrical Engineering Department, University of California, Los Angeles

[2] Martin Clark, ”MATLAB Central model: IEEE 802.11a WLAN PHY”

[3] Oskar Bexell, ”Antenna diversity gain in the wireless local area network standards HiperLAN/2 and IEEE 802.11a”, Lulea University of Technology, 2002

[4] Andreea Goldsmith, ”Wireless Communications”, Stanford University, Cambridge Univ. Press, 2005

[5] IEEE Std 802.11a-1999, ”Telecommunications and information exchange between systems. LANs and MANs. Specific requirements Part 11: Wireless LAN MAC and PHY specifications High-speed PHY in the 5 GHz Band”, 1999

[6] Ligia Chira, Tudor Palade, ”Adaptive Radio Techniques at 5 GHz”, *Acta Tehnica Napocensis*, Vol.46, No.2, Technical University of Cluj-Napoca, pp.1-6, Mediamira 2005

Ligia Chira, Tudor Palade
Technical University of Cluj-Napoca
Faculty of Electronics, Telecommunications and Information Theory
Communications Department
Address: Romania, Cluj-Napoca, 400027, 26, Baritiu street
E-mail: {Ligia.Chira,Tudor.Palade}@com.utcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 135-139

A Parallel FIFO Preflow Algorithm for the Minimum Flow Problem

Laura Ciupală, Eleonor Ciurea

Abstract: In this paper, we describe a parallel implementation of the sequential FIFO preflow algorithm for the minimum flow problem. The sequential algorithm was described by Ciurea and Ciupală (2004) in [8] and runs in $O(n^3)$ time. Our parallel algorithm runs in $O(n^2 \log n)$ time on a n- processors PRAM.

Keywords: Network flow; Network algorithms; Minimum flow problem; Parallel algorithms.

1 Introduction

The literature on network flow problem is extensive. Over the past 50 years researchers have made continuos improvements to algorithms for solving several classes of problems. From the late 1940s through the 1950s, researchers designed many of the fundamental algorithms for network flow, including methods for maximum flow and minimum cost flow problems. In the next decades, there are many research contributions concerning improving the computational complexity of network flow algorithms by using enhanced data structures, techniques of scaling the problem data etc.

There are many problems that occur in economy that can be reduced to minimum flow problems. Although it has its own applications, the minimum flow problem was not treated so often as the maximum flow and the minimum cost flow problem.

The minimum flow problem in a network can be solved in two phases:
(1) establishing a feasible flow, if there is one
(2) from a given feasible flow, establish the minimum flow

The problem of determining a feasible flow can be reduced to a maximum flow problem (for details see [1]).

For the second phase of the minimum flow problem there are three approaches:

1. using decreasing path algorithms (see [8], [9])
2. using preflow algorithms (see [8], [9])
3. finding a maximum flow from the sink node to the source node in the residual network (see [3], [6]).

The preflow algorithms for the minimum flow are more efficient than the decreasing path algorithms. In [8], Ciurea and Ciupală presented a generic preflow algorithm that runs in $O(n^2m)$ time and two special implementations of it: FIFO preflow algorithm that runs in $O(n^3)$ time and highest preflow algorithm that runs in $O(n^2\sqrt{m})$ time.

For the maximum flow problem there are several parallel preflow algorithm (see [11], [12], [16], [17]), but there is only one parallel algorithm for establishing a minimum flow (see [8]).

In this paper, we describe a parallel implementation of the FIFO preflow algorithm that runs in $O(n^2 \log n)$ time on a n- processors PRAM.

2 Notation and definition

We consider a capacitated network $G = (N, A, l, c, s, t)$ with a nonnegative capacity $c(i, j)$ and with a nonnegative lower bound $l(i, j)$ associated with each arc $(i, j) \in A$. We distinguish two special nodes in the network G : a source node s and a sink node t.

Let $N = \{1, 2, ..., n\}$, $|A| = m$ and $\overline{c} = \max\{c(i, j) \,|\, (i, j) \in A\}$.

A *flow* is a function $f : A \to R_+$ satisfying the next conditions:

$$f(i,N) - f(N,i) = \begin{cases} v, & i = s \\ 0, & i \neq s,t \\ -v & i = t \end{cases} \tag{1.a}$$

$$l(i, j) \leq f(i, j) \leq c(i, j),\ \forall (i, j) \in A, \tag{1.b}$$

for some $v \geq 0$, where

$$f(i,N) = \sum_{j\,|\,(i,j)\in A} f(i,j)$$

and

$$f(N,i) = \sum_{j\,|\,(j,i)\in A} f(j,i).$$

We refer to v as the *value* of the flow f.

The minimum flow problem is to determine a flow f for which v is minimized.

For the minimum flow problem, a *preflow* is a function $f : A \to \Re_+$ satisfying the next conditions:

$$\sum_y f(x,y) - \sum_y f(y,x) \leq 0,\ x \in N \setminus \{s,t\} \tag{2.a}$$

$$\ell(x,y) \leq f(x,y) \leq c(x,y). \tag{2.b}$$

Let f be a preflow. We define the *deficit* of a node $x \in N$ in the following manner:

$$e(x) = \sum_y f(x,y) - \sum_y f(y,x). \tag{3}$$

Thus, for the minimum flow problem, for any preflow f, we have $e(x) \leq 0,\ x \in N \setminus \{s,t\}$.

We say that a node $x \in N \setminus \{s,t\}$ is *active* if $e(x) < 0$ and *balanced* if $e(x) = 0$. We adopt the convention that the source node and the sink node are never active.

A preflow f for which

$$e(x) = 0,\ x \in N \setminus \{s,t\}$$

is a flow. Consequently, a flow is a particular case of preflow.

For the minimum flow problem, the *residual capacity* $r(i,j)$ of any arc $(i,j) \in A$, with respect to a given flow f, is given by $r(i,j) = c(j,i) - f(j,i) + f(i,j) - l(i,j)$. By convention, if $(j,i) \notin A$ then we add arc (j,i) to the set of arcs A and we set $l(j,i) = 0$ and $c(j,i) = 0$. The residual capacity of the arc (i,j) represents the maximum amount of flow from the node i to node j that can be cancelled. The network $G_f = (N, A_f)$ consisting only of the arcs with positive residual capacity is referred to as the *residual network* (with respect to preflow f).

In the residual network G_f, the *distance function* $d : N \to \mathcal{N}$ with respect to a given preflow f is a function from the set of nodes to the nonnegative integers. We say that a distance function is *valid* if it satisfies the following conditions:

$$d(s) = 0$$

$$d(j) \leq d(i) + 1, \text{ for every arc}(i,j) \in A_f.$$

We refer to $d(i)$ as the distance label of node i.

Lemma 2.1.[8] (a) *If the distance labels are valid, the distance label $d(i)$ is a lower bound on the length of the shortest directed path from node s to node i in the residual network.*

(b) *If $d(t) \geq n$, the residual network contains no directed path from the source node to the sink node.*

We say that the distance labels are *exact* if for each node $i, d(i)$ equals the length of the shortest path from node s to node i in the residual network.

We refer to an arc (i,j) from the residual network as an *admissible arc* if $d(j) = d(i) + 1$; otherwise it is *inadmissible*.

3 Parallel algorithm

The sequential FIFO preflow algorithm for the minimum flow problem developed in [8] is a specific implementation of the generic preflow algorithm for the minimum flow problem ([8]), that examines the active nodes in FIFO order. The active nodes are maintained in a queue. The node examinations are partitioned into different phases. The first phase consists of node examinations for those nodes that become active during initializations. The second phase consists of node examinations of all the nodes that are in the queue after the algorithm has examined the nodes in the first phase. The third phase consists of node examinations of all the nodes that are in the queue after the algorithm has examined the nodes in the second phase and so on.

The parallel FIFO preflow algorithm for the minimum flow problem, like the sequential algorithm, works in phases. But, it examines all nodes in a phases at a time.

Our parallel implementation of the FIFO preflow algorithm uses the following parallel pefix operations:

1. Given $q \leq n$ numbers $d(i_1), d(i_2), ..., d(i_q)$ compute the minimum of these numbers.
2. Given $q \leq n$ numbers $g(i_1), g(i_2), ..., g(i_q)$ compute the prefix sums $G(i_1) = g(i_1)$, $G(i_2) = g(i_1) + g(i_2), ...,$ $G(i_q) = g(i_1) + g(i_2) + ... + g(i_q)$.
3. Given a number e and $q \leq n$ numbers $G(i_1)$, $G(i_2), ...,$ $G(i_q)$ such that $G(i_1) \leq G(i_2) \leq ... \leq G(i_q) \geq e$ determine the minimum index w such that $G(i_w) \geq e$.

These operations can be performed in $O(\log n)$ using $n/\log n$ processors (for details see [2], [10]).

The parallel implementation of the FIFO preflow algorithm is the following:

Parallel FIFO preflow algorithm;
begin
let f be a feasible flow in network G;
compute the exact distance labels $d(\cdot)$ in the residual network G_f by
applying the BFS parallel algorithm from the source node s;
if t is not labeled **then** f is a minimum flow
else
begin
for each arc $(i,t) \in A$ **do in parallel** $f(i,t) := l(i,t)$;
$d(t) := n$;
while the network contains an active node **do**
begin
for all active nodes j **do in parallel**
begin
$k_j := 0$;
for all admissible arcs (i, j) **do in parallel**
begin
$k_j := k_j + 1$;
$g_j(k_j) := r(i, j)$;
end;
determine in parallel the prefix sums $G_j()$ of $g_j()$;
if $G_j(k_j) < -e(j)$ **then**
begin
for all admissible arcs (i, j) **do in parallel**
pull $r(i, j)$ units of flow from node j to node i;
determine in parallel $d(j) = \min\{d(i) \mid (i, j) \in A_f\} + 1$;
end
else begin
determine in parallel the minimum index w such that
$G_j(w) \geq -e(j)$;
for first $w-1$ admissible arcs (i, j) **do in parallel**
pull $r(i, j)$ units of flow from node j to node i;
let (i, j) be the w-th admissible arc entering in j;
pull $\min\{G_j(w) - e(j), r(i, j), \overline{r} + e(i)\}$ units

```
                    of flow from node j to node i
            end;
        end
        update in parallel the node deficits;
      end
    end
end.
```

To analyze the complexity of the parallel algorithm, we recall that the sequential algorithm performs $O(n^2)$ phases (see [8] for details). Since our parallel algorithm effectuates all node examinations in a phase at a time, its complexity is $O(n^2)$ times the time needed for the parallel examination of nodes. Since the sequential FIFO preflow algorithm for the minimum flow problem runs in $O(n^3)$ time, in order to obtain an optimal cost parallel algorithm, we cannot use more than $O(n)$ processors and we have assign the work to processor in such a way that most processors are busy most of the time. For this, we will use partial sum trees. We associate with each node j two partial sum trees: $in-tree(j)$, whose leaves correspond to the arcs entering into node j and $out-tree(j)$, whose leaves correspond to the arcs out-coming from node j. Using the partial sum trees, the algorithm can perform the operations of pulling flow and relabeling nodes that occur in the parallel examination of all nodes in a phase in $O(\log n)$ time using n processors. Thus, we have establish the following result:
Theorem 4.1. *On a PRAM with n processors, the parallel FIFO preflow algorithm runs in $O(n^2 \log n)$ time.*

4 Summary and Conclusions

In this paper, we developed a parallel implementation of the FIFO preflow algorithm for the minimum flow problem. Our parallel algorithm examines all active nodes in a phases at a time and runs in $O(n^2 \log n)$ time on a PRAM with n processors.

Further research ideas: parallel implementations might be used to speed up other sequential minimum flow algorithms and to solve more quickly other classes of network flow problems, for example minimum cost flow problems, dynamic flow problems etc.

References

[1] R. Ahuja, T. Magnanti and J. Orlin, *Network flows. Theory, algorithms and applications*, Prentice Hall, NJ, 1993.

[2] S. Akl, *The design and analysis of parallel algorithms*, Prentice Hall, NJ, 1989.

[3] J. Bang-Jensen, G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer-Verlag, London, 2001.

[4] L. Ciupală, "A scaling out-of-kilter algorithm for minimum cost flow", to appear in *Control and Cybernetics*,Vol. 34, no. 4, 2005.

[5] L. Ciupală, "About universal maximal dynamic flows", *Annals of University of Bucharest* Vol. 53, no.1, 115-124, 2004.

[6] L. Ciupală, E. Ciurea, "An algorithm for the minimum flow problem", *The Sixth International Conference of Economic Informatics*, 565-569, 2003.

[7] L. Ciupală, E. Ciurea, "An aproach of the minimum flow problem", *The Fifth International Symposium of Economic Informatics*, 786-790, 2001.

[8] E. Ciurea, L. Ciupală, "Sequential and parallel algorithms for minimum flows", *Journal of Applied Mathematics and Computing* Vol. 15, no.1-2, 53-78, 2004.

[9] E. Ciurea, L. Ciupală, "Algorithms for minimum flows", *Computer Science Journal of Moldova* Vol. 9, no.3(27), 275-290, 2001.

[10] E. Dekel, S. Sahani, "Binary trees and parallel scheduling algorithms", *IEEE Trans. Comput.* Vol. 10, 657-675, 1983.

[11] A. V. Goldberg, "Processor-efficient implementation of a maximum flow algorithm", *Information Processing Letters* Vol. 38, 179-185, 1991.

[12] A. V. Goldberg, R. E. Tarjan, "A Parallel Algorithm for Finding a Blocking Flow in an Acyclic Network", *Information Processing Letters* Vol. 31, 265-271, 1989.

[13] A. V. Goldberg, R. E. Tarjan, "A New Approach to the Maximum Flow Problem", *Journal of ACM* Vol. 35, 921-940, 1988.

[14] B. Hoppe, *Efficient dynamic network flow algorithms*, Ph.D Thesis, Cornell University, 1995.

[15] B. Hoppe, E. Tardos, "Polynomial time algorithms for some evacuation problems", *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discret Algorithms*, 433-441, 1994.

[16] Y. Shiloach, U. Vishkin, "An $O(n^2 \log n)$ parallel MAX-FLOW algorithm", *Journal of Algorithms* Vol. 3, 128-146, 1982.

[17] U. Vishkin, "A parallel blocking flow algorithm for acyclic networks", *Journal of Algorithms* Vol. 13, 489-501, 1992.

Laura Ciupală, Eleonor Ciurea
Transilvania University Braşov
Department of Computer Science
Address: Braşov, Iuliu Maniu St. 50
E-mail: laura_ciupala@yahoo.com, e.ciurea@unitbv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 140-145

A Programme Product for Solving Linear Optimization Problems

Moise Cocan

Abstract: The paper presents a new educational software product for learning simplex, dual simplex and revised simplex algorithms, from both the theoretical and applicative viewpoints. The developed software product is endowed with a particularly user-friendly interface. The applications illustrate these aspects.

Keywords: software product, dual simplex, revised simplex

1 Introduction

The Simplex application developed in Visual basic 6.0 offers the possibility of learning and applying solving modalities of linear programming problems by means of the simplex algorithm. The programme implements the three known variants of the simplex algorithm:

- two-phase simplex algorithm;
- dual simplex algorithm;
- revised simplex algorithm.

The most important facilities made available by the Simplex application are:

- solving of linear programming problems by the three known variants of the simplex algorithm: two-phase simplex, dual simplex and revised simplex;
- step-by-step simulation of the running of the algorithm designed for identifying the solutions of the problem;
- providing of a full report on the running of the linear programming problem solving algorithm;
- (verifying of the obtained solutions by matching against the data of the problem;
- visualisation of a geometrical interpretation of the problem in the 2D case, that is for two variable linear programming problems, the two-phase algorithm;
- automatic saving and loading of the data of the problem from files stored on the computer disk;
- providing a complete help facility, describing the utilisation of the application.

In order to successfully use the Simplex programme, the following hardware requirements need to be satisfied:

- Windowsl' 95/98 or NT on Intel platform;
- minimum 8 Mb RAM;
- VGA, SVGA or compatible video board (set on minimum 256 colours);
- approximately 10 Mb free HDD (hard disk) space;
- optionally a printer (for result printouts).

2 The model of the optimisation problems solvable with the proposed product

The used linear programming problems have the following general form:

$$\begin{cases} opt\,(c_1x_1 + c_2x_2 + \ldots + c_nx_n) \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \rho_1 b_1 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \rho_m b_m \\ x_i \geq 0, i = \overline{1,n} \end{cases} \tag{1}$$

3 Solving of the problems

In order to specify a problem in its entirety, the following data need to be entered: the number of variables and the number of restrictions; the coefficients of the objective function; the type of the problem (minimum or maximum problem); the coefficients of the restrictions as well as their free terms; the operator associated to each restriction individually ($<=, =, >=$).

In order to enter the data of a linear programming problem via the keyboard, one of the windows pulling the simplex algorithms is activated. The first step is to enter the number of variables as well as the number of restrictions of the problem, as in the figure below:

Figure 1: The number of variables and the number of restrictions

The type of the problem is specified (minimum or maximum problem):

Figure 2: The type of the problem

In order to complete the entering of the problem, further the coefficients of the objective function, the coefficients of the restrictions, the free terms, as well as the operators associated to the restrictions need to be entered (see figure 3).

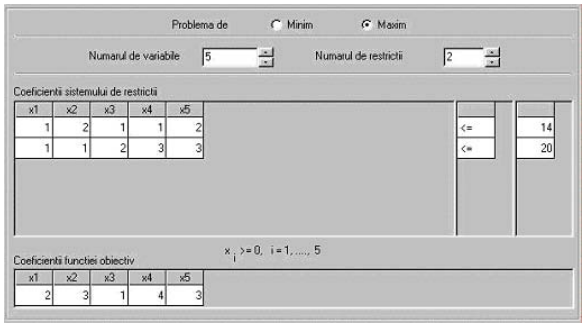


Figure 3: Specification of the data of the problem

If previously a problem has been entered manually and saved on the PC disk, this can be automatically loaded by pressing the Load (Încarca) key or by pressing F3.

A dialogue window as shown in figure 5 will open.

The three extensions used by the programme are:

- spx: for saved two-phase simplex problems;
- spd: for saved dual simplex problems;
- spr: for saved revised simplex problems.

The file format is identical for all the three extensions, thus regardless of the selected simplex window (two-phase, dual or revised) any file can be used to load a problem. The problem is saved in a file on the PC disk by pressing the Save (Salveaza) key or F2. A dialogue window will open as the one shown in figure 6, requiring a file name.

Figure 4: Load and Save Window

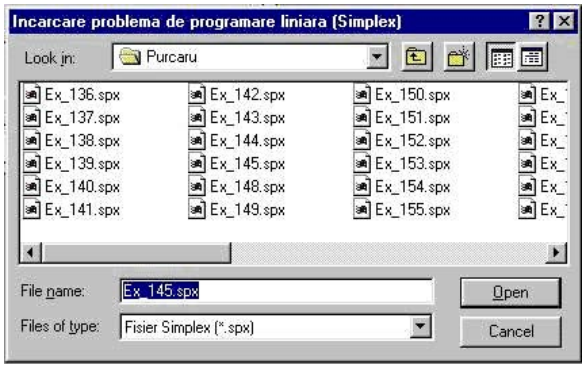


Figure 5: Automatic loading of a problem

Figure 6: Saving the data of a problem

In order to solve a linear programming problem entering (manually or from a file) of the problem data is required. Each window of the algorithm has at least two sections: Problem data and Algorithm running, as shown in figure 7.

Figure 7: Sections of the simplex window

For the two-phase simplex algorithm, the simulation table of the respective section is shown in figure 8.

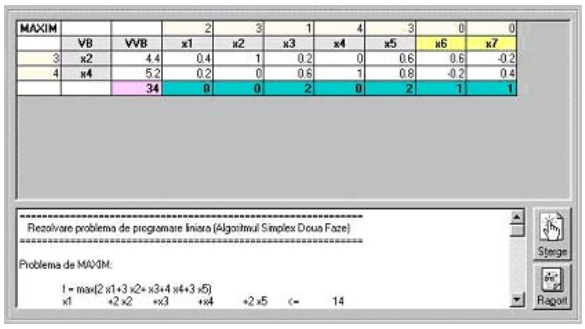


Figure 8: Simulation table of the two-phase simplex algorithm

In order to activate the running of the simplex algorithm, the Solve problem (Rezolva problema) key is pressed, or alternately F5. In the algorithm running section a simulation table will appear featuring the respective calculations (see figure 9).

MINIM			0	0	0	0	1	1	1
	VB	VVB	x1	x2	x3	x4	x5	x6	x7
1	x5	6	1	0	0	0	1	-1	0
0	x4	5	0.5	0.5	1	1	0	0.5	0
1	x7	10	1	0	-1	0	0	-1	1
		16	2	0	-1	0	0	-3	0

Figure 9: Simulation table

It can be noticed that the cells of the table are marked with different colours symbolising the contained type of information. The running of the algorithm can be stopped by pressing Esc or by activating the Close (Închide) key. The inquiry appears, whether the running of the algorithm should be cancelled or not.

4 Visualisation of the problem results

Upon completion of the algorithm the result is displayed in a dialogue window as shown in figure 10. The results can be a single optimum, a multiple optimum, infinite optimum or problem without programmes.

For a detailed report the Report (Raport) key will be pressed. The report window opens, wherefrom the solving detailed can be copied, saved or printed.

5 Geometrical interpretation of linear programming problems with two variables

The two-phase variant of the simplex algorithm offers for the category of two variable linear programming problems the possibility of graphic visualisation of the solving of the problem, which can support an intuitive verification of the obtained solutions, but most importantly visualises the domain of programmes given by the system of restrictions of the problem. Thus it suffices to load a two variable linear programming problem, solve it

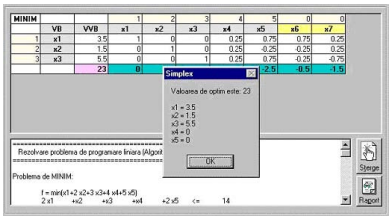


Figure 10: Result of a linear programming problem

and then activate the Geometrical interpretation (Interpretare grafica) section. By pressing the Graphic visualisation key the graphic visualisation window shown in figure 11 will open

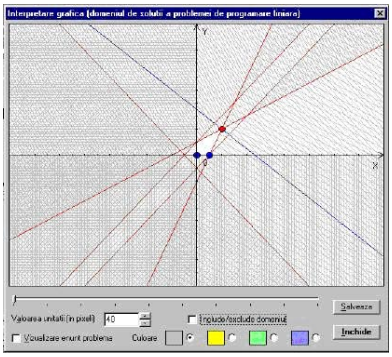


Figure 11: Graphic representation of the problem domain

6 Additional facilities

A first additional facility offered by the Simplex programme is the simultaneous solving of several problems, the working surface being a multi-window one (see figure 13).

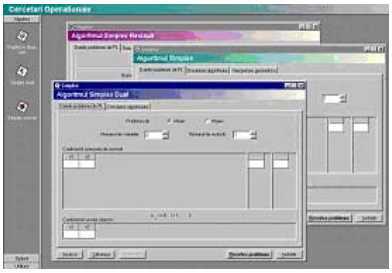


Figure 12: Multi-window in Simplex program

For rapid utilisation the programme offers a number of shortcut keys assigned to the commands. The table below lists the available shortcut keys: Key Action:
ESC Closes the active window
F1 Help
F2 Save problem
F3 Load problem
F5 Run simplex algorithm
F6 Verify obtained solutions.

7 Example

In figure 13 we present an example of linear programming problem.

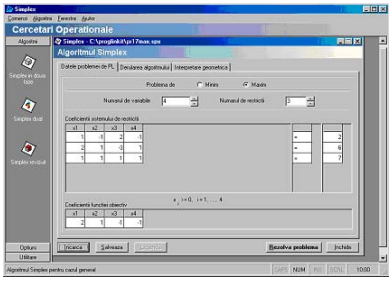


Figure 13: Linear programming problem

References

[1] M. Cocan, A. Vasilescu, *Programarea matematica folosind MS Excel Solver, Management Scientist, MatLab*, Editura Albastra, Cluj-Napoca, 1999.

[2] M. Cocan, *Modele, algoritmi si produse software în cercetarea operationala*, Editura Albastra, Cluj-Napoca, 2005.

[3] I. Purcaru, *Elemente de algebra si programare liniara*, Editura Stiintifica si Enciclopedica , Bucuresti, 1982.

[4] E. Marinescu, M. Stoian, *Research report*,Transilvania University of Brasov, 2000

Moise Cocan
Transilvania University Braşov
Department of Computer Science
Address: Iuliu Maniu 50, Braşov, Romania
E-mail: m.cocan@unitbv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 146-149

Using Centrality Indices in Ant Systems

Gloria Cerasela Crişan, Elena Nechita, Mihai Talmaciu, Bogdan Pătruţ

Abstract: Lately, much attention has been posited on evolutionary strategies that bring together self-organizing systems and nature selection inspired methods. Among these, Ant Colony Optimization algorithms have been suggested by the foraging behaviour of real ants. They can solve any optimization problem involving complex and heterogenous nodes, so these algorithms have been used to obtain solutions for many real-world problems. This paper presents some conclusions on introducing centrality indices as heuristics in Ant Systems.

Keywords: Ant Colony Optimization, centrality, swarm inteligence

1 Introduction

Swarm intelligence is a relatively new discipline that deals with the study of self-organizing processes both in nature and in artificial systems. Algorithms inspired by these models have been proposed to solve difficult computational problems.

A particularly successful research direction in swarm intelligence is Ant Colony Optimization (ACO), the main focus of which is on discrete optimization. ACO has been applied successfully to a large number of difficult problems including travelling salesman problem, the quadratic assignment problem, scheduling, vehicle routing.

The Traveling Salesman Problem (TSP) is a popular problem in the AI community, since it is simple to understand and very difficult to solve (NP-hard) at the same time. A salesman needs to complete a tour of a certain number of cities, using the most efficient path possible. He can travel from any city to any other city, but must visit each city once and only once.

In [4] the authors introduced Ant Systems (AS), the first ACO algorithms, using the frame of TSP. The simpliest AS works as follows: ants, placed in the vertices of the complete graph G=(V,E) determined by the cities, make a number of tours starting from a random city, depositing pheromones as they go. At the time t, when placed in city i, an ant k picks its next destination city j, according to the following probability:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{r \in A_k(t)} [\tau_{ir}(t)]^\alpha [\eta_{ir}]^\beta} & if\ j \in A_k(t) \\ 0 & otherwise \end{cases} \quad (1)$$

where $A_k(t)$ is the set of the permitted nodes for the k-th ant at the moment t. The elements of this probability are the following:
i. The visibility $\eta_{ij} = \frac{1}{d_{ij}}$, d_{ij} being the distance between i and j. The closest i and j are, the bigger the visibility is.
ii. The trail $\tau_{ij}(t)$ models the quantity of pheromones on the edge (i,j) at the moment t.
iii. α and β are two parameters that measure the importance of the pheromone trail versus visibility.

After a tour is completed, an ant deposits a certain amount of pheromone on each edge of the graph, depending on how much the ant travelled during its tour. Shortest tours lead to more pheromone being deposited. A certain amount of pheromone will also decay, causing older solutions to fade away and be replaced by new ones.

AS and subsequent versions of them do not necessarily find the optimal solution, but are effective in finding good solutions in a reasonable number of iterations.

2 Introducing centrality indices

The idea behind the expression of $p_{ij}^k(t)$ is that this probability grows if j is closer to i and if the pheromone trail on the edge (i,j) has been previously enforced. Starting from this natural approach, we thought of introducing some centrality indices [1] in formula (1). The intuition about a centrality index is that it denotes an order of importance on the vertices of a graph, by assigning real values to them.

Using this supplementary information, a new element of control can be used to improve the search of the optimal solution.

Three centrality indices have been used:
i. **The centroid value.** The centroid value of a vertex $x \in V$ measures the advantage of the vertex x compared to other vertices and is defined as

$$cen(x) = min\{f(x,y), y \in V - x\} \quad (2)$$

where $f(x,y) = g_x(y) - g_y(x)$, $g_x(y) = |\{z \in V : d(x,z) < d(y,z)\}|$.
ii. **The degree.** The centrality degree of a vertex is determined by the number of its neighbors. The degree of a vertex $x \in V$ (reffereed in [2] as closeness) is

$$deg(x) = sum_{y \in V} d(x,y) \quad (3)$$

iii. **The eccentricity.** The eccentricity of a vertex $x \in V$ is the maximum distance to any other vertex $y \in V$:

$$ecc(x) = max\{d(x,y), y \in V\} \quad (4)$$

3 Results of the experiments

The three centrality-based AS have been tested on three benchmark problems:
eil51 (in TSPLIB, http://iwr.uni_heidelberg.de/groups/comopt/software/TSPLIB95/tsp/)
wi29 and xqf131 (from http://www.tsp.gatech.edu)

Let *ind* be any of the centrality indices (2), (3), (4). The values are computed for every $x \in V$ and ascendingly sorted in the array *sind* . For each of the three indices, two heuristics have been implemented. When an ant is placed in node *i*, the node *j* to move to is chosen according to the visibility η_{ij} , that is defined as follows:
A. The inverse of

$$d_{ij} * | ind(j) |$$

and in this case, the choice of node *j* is guided by the "importance" of node *j*, which is reflected in the absolute value of *ind(j)*.
B. The inverse of

$$d_{ij} * | sind(i) - sind(j) |$$

and in this case the distance (in the array *sind*) between nodes *i* and *j* is taken into account.

The results of the experiments are presented into the following tables.

Problem	Nodes	Optimum	Time (s)	A Method	B Method
wi29	29	27603.0	3	min: 37.49 average: 40.66	min: 75.59 average: 79.08
eil51	51	429.98	10	min: 18.61 average: 21.40	min: 49.62 average: 54.89
xqf131	131	564.0	170	min: 27.12 average: 29.18	min: 99.31 average: 102.53

Table 1. Results for the *cen* index

Problem	Nodes	Optimum	Time (s)	A Method	B Method
wi29	29	27603.0	3	min: 27.23 average: 29.48	min: 44.99 average: 49.95
eil51	51	429.98	10	min: 6.57 average: 7.76	min: 66.32 average: 69.54
xqf131	131	564.0	170	min: 8.91 average: 9.68	min: 68.56 average: 71.83

Table 2. Results for the *deg* index

Problem	Nodes	Optimum	Time (s)	A Method	B Method
wi29	29	27603.0	3	min: 27.66 average: 29.90	min: 32.45 average: 34.56
eil51	51	429.98	10	min: 7.03 average: 7.81	min: 75.17 average: 80.46
xqf131	131	564.0	170	min: 10.20 average: 10.83	min: 83.21 average: 91.64

Table 3. Results for the *ecc* index

The values in the columns corresponding to heuristics A and B represent the minimum (first value) and the mean (second value, over ten executions) of the distances between the solutions obtained with the AS and the optimum solutions.

4 Conclusions

As one can observe, the best results have been obtained for the large problems, with method A, while method B leads to solutions far away from the optimal one. Future work on this approach implies tests of method A on even larger problems also considering the structure of these problems. On highly different problems, the method could lead to interesting results, depending on the graph topology.

References

[1] U. Brandes, T. Erlebach, (Eds.) *Network Analysis*, LNCS 3418, Springer-Verlag, 2005

[2] T. McCallum, *Understanding how knowledge is exploited in Ant algorithms*, PhD Thesis, 2006

[3] M. Dorigo, G. Di Caro, "Ant Algorithms for Discrete Optimization", *Artificial Life*, Vol.5, No. 3, pp. 137-172, 1999

[4] M. Dorigo, V. Maniezzo, A. Colorni, "The Ant System: Optimzation by a colony of cooperating agents, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, (26)1, pp. 29-41, 1996

[5] W. Gutjahr, "ACO Algorithms with Guranteed Convergence to the Optimal Solution", *Information Processing Letters*, No. 82, pp. 145-153, 2002

[6] T. Stutzle, H. Hoos, "MAX-MIN Ant System", *Future Generation Computer System*, No. 16, pp 889-914, 2000

[7] M. Zlochin, M. Birattari, N. Meuleau, M. Dorigo, "Model-Based Search for combinatorial Optimization: A Critical Survey" - *Annals of Operations Research*, No. 131, pp. 376-395, 2004

Gloria Cerasela Crişan, Elena Nechita
Mihai Talmaciu, Pătruţ Bogdan
University of Bacău
Department of Mathematics and Informatics
Address: 8 Spiru Haret str., 600114 Bacău, ROMANIA
E-mail: {ceraselacrisan,elenechita,mihaitalmaciu}@yahoo.com, bogdan@edusoft.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 150-155

Telemonitoring System for Complex Telemedicine Services

Hariton Costin, Cristian Rotariu, Bogdan Dionisie, Roxana Ciofea, Sorin Puşcoci

Abstract: In spite of decreased mortality, coronary artery disease still remains the leading cause of death almost all over the world. The existence of silent myocardial ischemia emphasizes the need for monitoring of the asymptotic patient. Extended patient monitoring during normal activity has become increasingly important as a standard preventive cardiological procedure for detection of cardiac arrhythmias, transient ischemic episodes and silent myocardial ischemia. Existing holter devices mostly record “24 hour activity" and then perform off-line record analysis, so they are not real-time. A telemonitoring network devoted to medical teleservices will enable the implementation of complex medical teleservices (teleconsulations, telemonitoring, homecare, urgency medicine, etc.) for a broader range of patients and medical professionals, mainly for family doctors and those people living in rural or isolated regions. Thus, a multimedia, scalable network, based on modern IT&C paradigms, will result. A first attempt for real-time electrocardiogram (ECG) acquisition, internet transmission and local analysis was already successfully done for patients monitoring.

Keywords: cardiology, telemedicine, telemonitoring

1 Introduction

In spite of decreased mortality, coronary artery disease still remains the leading cause of death almost all over the world. The existence of silent myocardial ischemia emphasizes the need for monitoring of the asymptotic patient. Extended patient monitoring during normal activity has become increasingly important as a standard preventive cardiological procedure for detection of cardiac arrhythmias, transient ischemic episodes and silent myocardial ischemia. Existing holter devices mostly record “24 hour activity” and then perform off-line record analysis, so they are not real-time.

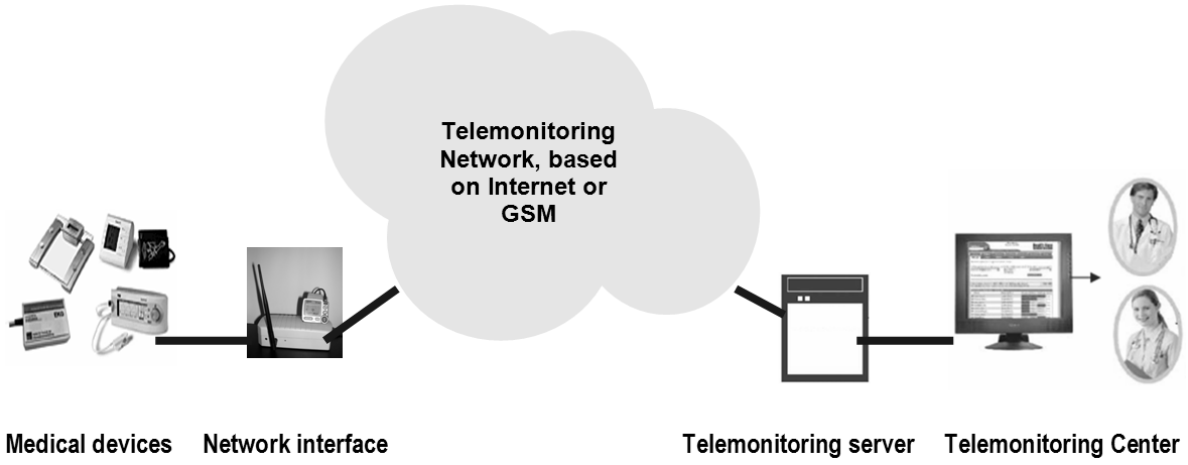


Figure 1: Medical telemonitoring network - general structure

The task may also be achieved by *telemedicine* (enabling medical information-exchange as the support to distant-decision-making) and *telemonitoring* (enabling simultaneous distant-monitoring of a patient and his vital functions), both having many advantages over traditional practice. A telemonitoring network (Fig.1) devoted to medical teleservices, will enable the implementation of complex medical teleservices for a broader range of patients and medical professionals, mainly for family doctors and those people living in rural or isolated regions.

Doctors can receive information that has a longer time span than a patient’s normal stay in a hospital and this information has great long-term effects on home health care, including reduced expenses for health care.

Physicians also have more accessibility to experts, allowing the physician to obtain information on diseases and provide the best health care available. Moreover, patients can thus save time, money and comfort.

As for patient monitoring, *we propose the development of a flexible environment based on an acquisition module and an embedded system for real-time biosignals processing and transmission through Internet, GPRS/3G (mobile telephony) or radio networks already existing in each Romanian county*.

2 Materials and methods

Our telemedicine module is based on an ECG / biosignal acquisition module and an embedded system, for real-time signal processing and transmission through Internet (Figure 2). The telemonitoring system is built by using custom developed hardware, open-source and application software.

For instance, the monitoring device could be used either for acquisition of anomalous ECG sequences (e.g. with arrhythmic events, ST segment deviation, etc.) and storing to a compact flash memory, as a warning device during normal activity, or an exercise stress test.

The hardware part is mainly based on biosignal amplifiers, an autocalibrating 16-bit analog I/O PC/104 module and an embedded Internet interface subsystem (MOPS/520). The device has as features: real-time ECG / biosignal acquisition and processing, executes the operatorŠs commands, monitors the systemŠs overall performance, acts in emergency situations, and aids the diagnostic.

In the following, we refer to the ECG acquisition, transmission and analysis application, already achieved by our team.

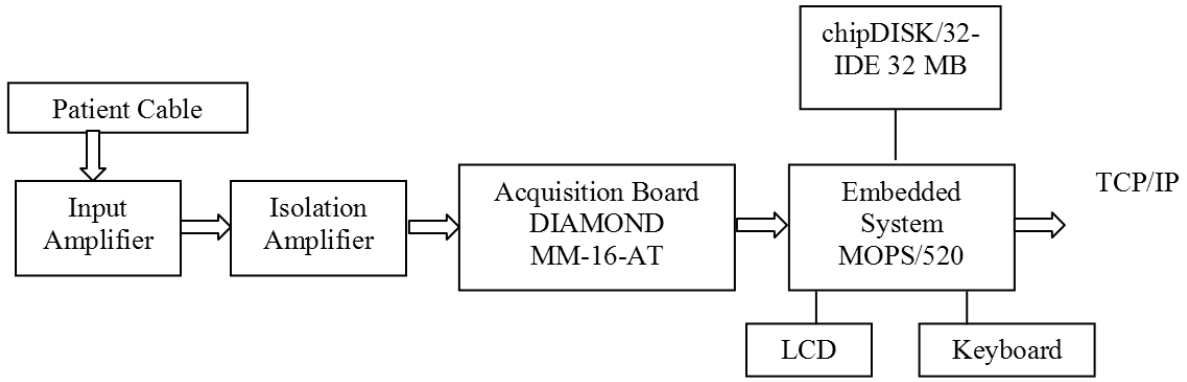


Figure 2: The biosignal / ECG monitoring unit

2.1 The ECG monitoring unit

The 12-leads ECG amplifier has for each channel a gain of 1000, is AC coupled and has a band limited to 0.05 - 150 Hz. The high common mode rejection ($> 100dB$), high input impedance ($> 100M\Omega$), the fully floating, isolated and defibrillation protected patient input are other features of the ECG amplifier.

The biosignal / ECG acquisition module is built around Diamond-MM-16-AT, a PC/104 expansion board offering a full feature set of data acquisition capabilities. It is used in any PC-compatible embedded computer with a PC/104 (ISA-bus) expansion connector. Its key features include: 16 single-ended / 8 differential and autocalibrated inputs, 16-bit A/D resolution, 100 KHz maximum A/D sampling rate, programmable input ranges with maximum range of +/-10V, 4 optional analog outputs, user-programmable output ranges, 8 dedicated digital inputs and outputs, TTL compatible.

The Internet interface MOPS/520 is based on a microcontroller (32-bit AMD5x86 CPU) that runs at clock speeds of 133 MHz. The system integrate the complete functionality of motherboard and include: CPU, system BIOS, up to 64 MB SDRAM, keyboard-controller, and real-time clock. Additional peripheral functions include: 4 serial, one parallel and 2 USB ports, IDE-hard disk interface, Ethernet access and CAN bus interface.

The used display is a LCD Seiko 628-G321 having dot pixels 320 x 240.

To make a simple software implementation, we choose to use the standard TCP/IP network protocol as the link provider, a scalable and economically feasible tool.

For DAQ applications in real-time, such as ours, one must use real time (RT), multitasking operating systems. A modern and economic solution is to choose an open source (free) RT-OS, such as RT-Linux. It is comprised of a small RT kernel which runs: (i) a C/C++ RT process at top priority, and (ii) the standard Linux kernel as a fully preemptable low priority task. High speed (low interrupt latency) and predictable timing are achieved by limiting the RT process to functions that are essential to real time.

2.2 ECG acquisition and processing

The most important ECG phases for morphological analysis are [1]:

- *P-wave* (representing contraction of the atria);
- *QRS complex* (representing contraction of the ventricles);
- *T-wave* (representing the recovery of the ventricles).

Typical ECG processing algorithms consist of the following steps:

a. Initialization – Used to determine initial signal and timing thresholds, positive/negative peak determination, automatic gain control, etc.

b. Filtering - This is performed first as analog filter on ECG amplifier board, and then as digital filter on acquisition board. In addition, a 50 Hz notch filter is used to reduce power line interference.

c. QRS complex detection - Reliable detection of R-peak is crucial for morphological analysis [3].

d. Baseline correction - Compensates for low-frequency ECG baseline drift.

e. ST segment detection [6].

2.3 Detection of QRS complexes

An adaptive thresholding technique with searchback serves as the primary method for QRS detection. The thresholds are based on the most recently detected signal and noise levels to react to changes in the patient's heart rate, as well as to signal and noise levels.

The QRS complex is the most significant feature in the ECG signal. Being characterized by sharp slopes, its duration is about 70 - 130 msec and its energy spectrum is mostly between 1 and 40 Hz. The input of the QRS detector is the digital ECG signal, sampled at 250 Hz and quantized with 12 bits/sample by A/D converter. The outputs are the limits of the QRS complex (QRS_{on} and QRS_{off}), the location of the R wave, and location of the QRS peaks and notches (if they exist) of every beat (complex) [4] [5]. The QRS detection algorithm consists of three steps: (1) coarse QRS limits determination; (2) peaks and notches determination, and (3) exact limits determination.

2.4 ST segment analysis

The ST-segment begins 40 milliseconds after the R-peak in the event the heart rate is more than 100 bpm, or 60 milliseconds after the R-peak otherwise. ST-segment has normally a predefined length of 160 milliseconds. The normal ST-segment template is constructed for each patient as the average of the first ten normal ST-segments. Baseline drift is compensated according to the slope between the isoelectric levels of the two beats. Standard annotated databases, such as the European ST-T Database and the MIT/BIH Arrhythmia Database, provide means for algorithm evaluation. In order to compute ST-segment length, a T wave detector must be implemented [2].

2.5 Data compression and error rate

Experiments revealed the necessity for data compression, in order to make a real-time ECG transmission. We used Linux Gzip programme, that yields about 2:1 average compression ratio by means of Lempel Ziv algorithm. Table 1 presents results obtained for a resolution of 12 bits/sample and 250 Hz sampling rate, for a 3 leads ECG. Thus, only 6 KB/s bit rate was enough for a real-time 3 leads ECG transmission!

The quality of data transmission was evaluated by computing PRD (percentage rate of distortion, a kind of root mean square), according to formula below. Table 2 shows a PRD level under 10%, a value accepted by clinicians for expressing a correct diagnosis.

Table 1: Compression results for different sizes of TCP/IP buffer

Time (sec.)	Uncompressed size (bytes)	Compressed size (bytes)	Compression ratio (%)
10	60000	26592	55.7
9	54000	24125	55.3
8	48000	21681	54.9
7	42000	19027	54.8
6	36000	16541	54.1
5	30000	14061	53.2
4	24000	11502	52.2
3	18000	8904	50.7
2	12000	6123	49.2
1	6000	3205	47.0

Table 2: Error rate for different signals from MIT/BIH Arrhythmia Database

$$PRD = \sqrt{\frac{\sum_{n=1}^{N}[x(n)-\tilde{x}(n)]^2}{\sum_{n=1}^{N}x^2(n)}} \times 100$$

Nr.	Signal	Time (min)	PRD
1	No. 100	1	6.2
		2	6.9
		3	7.2
2	No. 201	1	6.3
		2	7.1
		3	7.9
3	No. 107	1	2.1
		2	2.2
		3	2.4

3 Results and discussion

The whole telemonitoring system acts as a client-server application. The server module includes: a database server (using MySQL and open sources for server procedures, tables, restrictions coming from “client” application); an administration/control module that supervises general dataflow; an access/security module; a parameters configuration module a.s.o. Also, it uses HTML and HTTP to send most up to date information on heart care to clients.

The client module comprises the software working on the expert’s computer. It is implemented by using Java applets and has the following facilities: GUI (Graphic User Interface) for ECG monitoring (Fig. 3); displays the patientŠs ECG in real-time and the extracted ECG segments data; communicates the experts’ commands (e.g. remote selection of the ECG lead) and medical decisions to the physician/patient. Also, some off-line processing algorithms are implemented, such as: advanced filtering; morphologic ECG analysis (intervals, amplitudes, electrical axes), average complexes with measurement reference markings; heart rate variability analysis, etc.

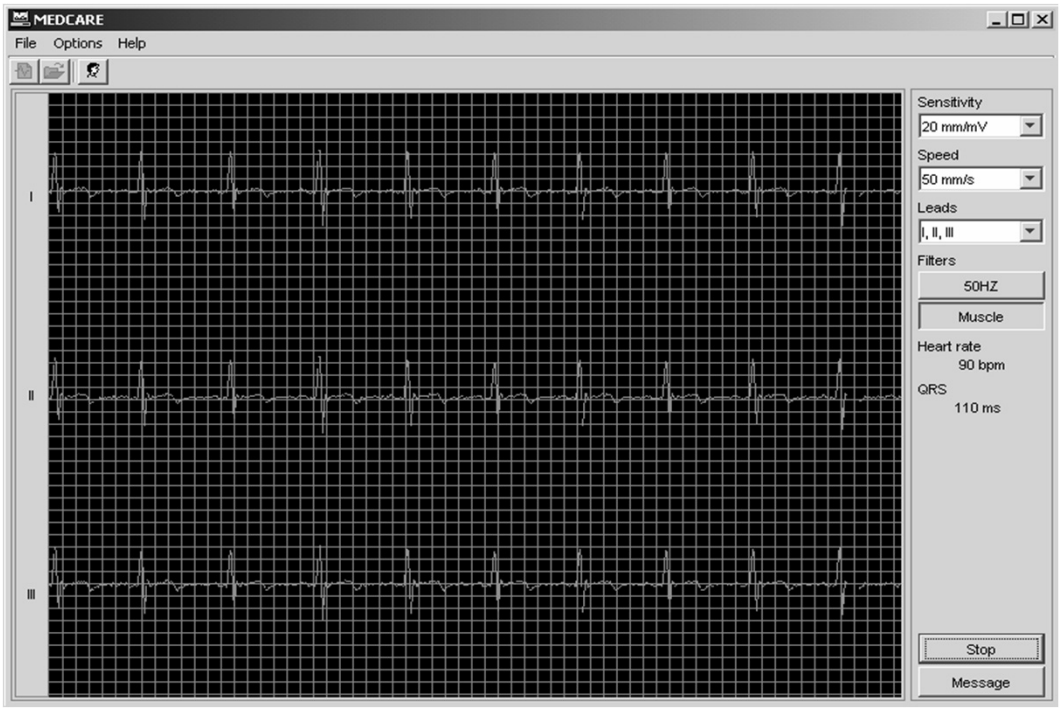


Figure 3: User interface for 3 leads ECG acquisition and analysis

We designed and prototyped the monitoring unit for acquisition and real-time ECG processing, the software implementation for the Internet connectivity (the embedded TCP/IP subsystem), and the software for displaying ECG information on the medical doctor's computer. The average reconstruction error of the ECG signal is about 4.6%. We also tested various algorithms for morphologic ECG analysis with good results on MIT/BIH Arrhythmia Database (Table 3).

Table 3: Morphological analysis of a 12 leads ECG

	Amplitude [μV]										Slope [$\mu V/s$]	Duration (ms)		
Lead	P+	P-	Q	R	S	ST20	ST60	ST80	T+	T-	ST	Q	R	S
I	171	0	-76	877	-130	-11	-3	-3	320	0	250	18	42	24
II	248	0	-125	1302	-205	-17	-8	-8	474	0	750	18	42	24
III	77	0	-49	425	-76	-6	-5	-5	154	0	500	12	44	24
aVR	0	-209	0	100	-192	-14	5	5	0	-397	-500	-	18	42
aVL	47	0	0	227	-28	-3	1	1	84	0	0	-	38	26
aVF	163	0	-87	864	-140	-11	-6	-6	314	0	750	18	42	24
V1	80	0	-46	451	-72	-6	-3	-3	160	0	250	14	44	24
V2	168	0	-88	908	-143	-12	-5	-4	327	0	750	18	42	24
V3	242	0	-126	1283	-205	-17	-9	-7	463	0	750	18	42	24
V4	375	0	-185	1945	-303	-24	-9	-9	712	0	1000	18	42	24
V5	247	0	-124	1305	-204	-17	-8	-7	477	0	500	18	42	24
V6	182	0	-79	918	-133	-11	-4	-3	338	0	500	18	42	24

Heart Rate	60 bpn
P Dur.	90 ms
PR Int.	164 ms
QRS Dur.	84 ms
QT Int.	372 ms
QTc Int.	372 ms
P Axis	43°
QRS Axis	43°
T Axis	44°

4 Summary and Conclusions

Real time personal ECG monitoring, as an important application of telemonitoring system, requires devices with high peak performance and low power consumption. High performance of RT-Linux development environment allows high speed multitasking procedures and real time signal processing. The proposed system could be used as a warning system (Holter-type) for monitoring of arrhythmia or ischemia during normal activity or physical exercise. In addition to monitoring of physiological signals, we plan to use the proposed environment for development of a high performance user interface. New user inputs, including correlates of

the user's physiological and emotional states could significantly improve human-computer interface and interaction. Many algorithms for ECG analysis have already been tested with very good results. Moreover, our

monitoring system is general enough to enable a wide range of biosignals monitoring and analysis, e.g. ECG, EEG, EMG a.s.o.

ECG tele-monitoring of a patient in real time, according to our project, has as main feature the analysis and transmission of the patients' bio-signals through the Internet, so that experts in cardiology could make the right diagnostic. So, by using the existing web-based and embedded technologies, the quality of medical decision in tele-healthcare and emergency medical services systems can be significantly improved.

Acknowledgements

This work is supported by a grant from the Ministry of Education and Research, within CEEX programme (www.mct-excelenta.ro), contract No. 604/645/21.10.2005.

References

[1] A. Cohen, *"Biomedical signal processing"*, in A. Prochazka *et al.* (eds.), *Signal Analysis and Prediction I*, EURASIP, ICT Press, Prague, 1997

[2] P. Laguna *et al.*, *"New algorithm for QT interval analysis in 24-hours holter ECG: performance and applications"*, Med. Biol. Eng. Comput.,Vol. 28, pp. 67-73, January 1990.

[3] J. Pan, W.J. Tompkins, *"A real-time QRS detection algorithm"*, IEEE Transactions on Biomedical Engineering, Vol. 32, No. 3, pp. 230-236, March 1985.

[4] D.L. Rollins *et al.*, *"A telemetry system for the study of spontaneous cardiac arrhythmias"*, IEEE Transactions on Biomedical Engineering, Vol. 47, No. 7, pp. 887-892, July 2000.

[5] A. Ruha *et al.*, *"A real-time microprocessor QRS detector system with a 1-ms timing accuracy for the measurement of ambulatory HRV"*, IEEE Transactions on Biomedical Engineering, Vol. 44, No. 3, pp. 159-167, 1997.

[6] V. Subramanian, *"Clinical and research applications of ambulatory Holter ST-segment and heart rate monitoring"*, The American Journal of Cardiology, Vol. 58, No. 4, pp. 11B-20B, 1986.

[7] European Commission, Information Society Technologies Directorate-General, *"Resource Book of e-Health Projects Ű 6th R&D Framework Programme, 2002-2006"*.

[8] European Commission, Information Society Technologies Directorate-General, *"Applications Relating to Health Ű 5th R&D Framework Programme, 1998-2002"*.

Hariton Costin, Cristian Rotariu, Bogdan Dionisie, Roxana Ciofea
"Gr. T. Popa" Univ. of Medicine and Pharmacy, Iasi
Faculty of Medical Bioengineering
Address: 16 Universitatii St., 700115, Iasi, Romania
E-mail: hcostin@gmail.com

Sorin Puşcoci
National Institute of Studies and Researches in Communications
Bucharest, Romania

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 156-162

Medical Image Analysis and Representation using a Fuzzy and Rule-Based Hybrid Approach

Hariton Costin, Cristian Rotariu

Abstract: When missing, ambiguous or distorted data is available in digital image processing, soft computing (e.g. fuzzy logic, neural networks, evolutionary computation) has proved to yield promising results. In the field of biomedical image analysis, when information has a strong structural character, many methods of artificial intelligence are well suited for knowledge discovery, representation and processing. Fuzzy logic acts as an unified framework for representing and processing both numerical and symbolic information, as well as structural information constituted mainly by spatial relationships in biomedical imaging. This paper describes the use of fuzzy logic at low level to a higher level (e.g. model based structural pattern recognition and scene understanding). Applications are for segmentation of brain structures in magnetic resonance (MR) and CT (computer tomography) images, based both on atlas and real data. Promising results show the superiority of this knowledge-based approach over best traditional techniques in terms of segmentation errors.

Keywords: medical imaging, fuzzy logic, image representation and segmentation, knowledge-based systems.

1 Introduction

The information to handle in medical images is often *heterogeneous*. For instance, when planning a surgical operation, necessary images can be: anatomical images (provided by MRI or CT), angiographic images (MRA, spiral CT, etc.), or functional images (PET, functional MRI). Another cause of heterogeneity comes from the need of expert knowledge related to the problem at hand. This knowledge can be expressed either in ”iconic” terms, under the form of atlases, or in “symbolic” terms, under the form of linguistic expressions or rules.

Imprecision in medical image information is due to several factors, ranging from the observed phenomenon to the algorithms' precision. A soft transition between healthy and pathological tissues is surely a cause of imprecision inherent to the nature of the observed objects. Also, if tissues have similar characteristics, images that represent these characteristics will poorly discriminate these tissues. For instance, in MRI several important cranial tumors have long T_1 and T_2, similar to normal brain. So, it may be difficult to distinguish tumor margins from edematous normal brain. Without MR contrast agents (Gd-DTPA), in at least 10% of cases tumor contours remain indeterminate [7]. This will result in an uncertainty on the belonging of a pixel to one or the other tissue.

The partial volume effect (the presence of several tissues in one pixel or voxel) belongs also to this type of spatial imprecision. Other image imperfections can be caused by numerical reconstruction algorithms in computed imaging. One example is the Gibbs effect that may appear in MRI around sharp transitions. At the processing level, imprecision is often induced by the chosen algorithms.

In this context, the *theory of fuzzy sets* appears particularly interesting and useful, as it provides a good theoretical basis to represent imprecision of the information and it constitutes an unified framework for representing and processing both numerical and symbolic information, as well as structural information. Therefore, this theory can achieve tasks at several levels, from low level (e.g. gray level based classification) to high level (e.g. model based structural pattern recognition and scene interpretation).

Proceeding from the grounds of artificial intelligence and soft computing, for instance from [6][14], a number of authors have used these techniques to aid in the analysis of medical images, e.g. [4][5][9][11][13].

2 Representation of Image Structures

The most common use of fuzzy sets in medical image processing at low level is for classification. We assign to each pixel or voxel in the image a membership degree to a class. Fuzzy sets can then be considered from two points of view. In the first one, a membership function is a function μ_i from the space S on which the image is defined to [0, 1]. The value $\mu_i(x)$ denotes the membership degree of x ($x \in S$) to the class i. In the second one, a membership function is defined as a function μ'_i from a space of attributes A into [0, 1]. At numerical level, such

attributes are typically the gray levels. The value $\mu_i'(g)$ represents the degree to which a gray level g supports the membership to the class i. We have $\mu_i(x) = \mu_i'[g(x)]$, where $g(x)$ denotes the gray level of x.

Such model explicitly represents imprecision in the imagistic information and as well as possible ambiguity between classes. For instance, a pixel or voxel affected by partial volume effect is characterized by its partial belonging to at least two different tissues or classes, i.e. by non zero membership values to several classes.

The partial membership to the pathology may be estimated according to the methods shown in [10].

Estimation and learning of membership functions is a difficult task. Some methods are based on the minimization of some criteria, the most used being the fuzzy *C*-means algorithm. Another class of methods relies on probability■possibility transformations, while other techniques are based on statistical information also by minimizing some criteria (e.g. [1][3]).

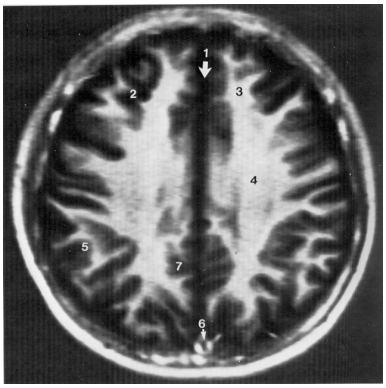


Figure 1. Axial MR 2D image of a brain: 1-interhemispheric fissure (IHF), 2-superior frontal sulcus (SFS), 3-superior frontal girus (SFG), 4-centrum semiovale (CS), 5-superior parietal lobule (SPL),6-superior sagital sinus (SSS),7-parieto-occipital sulcus (POS)[7](courtesy of Elsevier Publ.

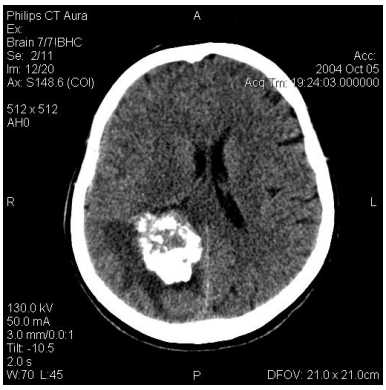


Figure 2. Axial CT image of a patient brain showing a tumoral lesion and a peritumoral edema

We now consider an object or structure in an image as a fuzzy set, which actually is a fuzzy subset of the image space S, to which a semantic meaning can be attached. This provides a higher level of representation than the pixel-based representations used in the previous model. As illustrative examples, two slices of brain images are shown in Figure 1 [7] and Figure 2 (a CT scan). Figure 1 is obtained using a T1 weighted acquisition in magnetic resonance imaging (MRI), and note the complementary representation of structures 3 and 4 (in whiter pixels).

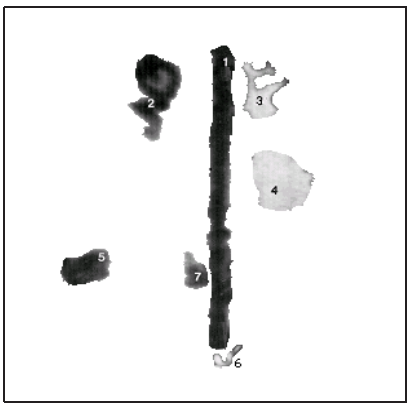


Figure 3. The 7 fuzzy objects representing internal brain structures of the image in Figure 1 (membership values rank between 0 and 1, from black to wihite.

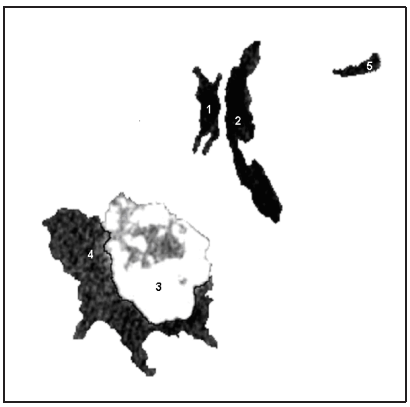


Figure 4. The similar segmentation as in Fig.3, for CT image in Fig.2; superposition of the objects: peritumoral edema(object 4); tumoral lesion(3); right lateral ventricle (1); left lateral ventricle (2); cortical ditch (5).

Some internal structures are represented through segmentation in Figure 3 and Figure 4 respectively, as iconic fuzzy sets (membership degrees are depicted using gray levels). The use of fuzzy sets may represent different types of imprecision, either on the boundary of the objects (due, e.g., to partial volume effect or to the spatial resolution), or on the individual variability of these structures.

Several operations have been defined on fuzzy objects, since the works of Zadeh [14] on set operations, and of Rosenfeld on geometrical operations [12]. As geometrical operations we use *area* and *perimeter* of a fuzzy object.

Table I shows examples of fuzzy areas and fuzzy perimeters for the objects in Fig. 3. For each fuzzy object, the cardinality of its support (i.e. the number of points having a strictly positive membership value), its fuzzy area, and the perimeter in 8■-connectivity are computed (in pixels). Such geometrical features can typically be used in shape recognition, where geometrical measures of the objects are often taken into account.

TABLE I. The areas and perimeters of the fuzzy, fuzzy objects in Figure 3

Fuzzy object	Size of support	Fuzzy area	Fuzzy perimeter
1-IHF	7683	3726	655
2-SFS	3413	1628	270
3-SFG	1872	854	254
4-CS	3593	1709	198
5-SPL	1953	829	147
6-SSS	412	176	92
7-POS	934	405	109

3 Representation of Structural Information

The main information contained in the images consists of *properties* of the objects and of *relationships* between objects, both being used for pattern recognition and scene interpretation purposes. Relationships between objects are particularly important since they carry structural information about the scene, by specifying the spatial arrangements between objects. These relations highly support structural recognition based on models like anatomical atlases.

We distinguish two types of relationships for representing structural information: the first one corresponds to relations that are *well defined* in the crisp case, and the second one to relations that are *vague* even in the crisp case. We will extend relationships of the first type to fuzzy objects, and illustrate this construction on two examples: *adjacency* and *distances*. Fuzzy concepts are powerful for defining relationships of the second type, even on crisp objects. The *relative position* is a good illustration.

3.1 Extending Crisp Relationships to Fuzzy Sets

A powerful approach for defining a fuzzy relation from the crisp one consists in translating binary equations into their fuzzy equivalent: intersection is replaced by a t-norm, union by a t-conorm, sets by membership functions, etc. Examples are found for defining fuzzy morphology [2], fuzzy inclusion, etc.

Adjacency is an example of spatial relationship that carries strong information about the structure of the image. In the crisp discrete case, two image regions X and Y are adjacent if

$$X \cap Y = \emptyset \text{ and } \exists x \in X, \exists y \in Y : n_c(x,y) \tag{1}$$

where $n_c(x,y)$ is the Boolean variable stating that x and y are neighbors in the sense of the discrete $c-$connectivity.

The extension of this definition involves the definitions of a degree of intersection $\mu_{int}(\mu,\nu)$ between two fuzzy sets μ and ν defined on S, a degree of non-intersection, $\mu_{\neg int}(\mu,\nu)$ and a degree of neighborhood n_{xy} between two points x and y of S. The definition for *fuzzy adjacency between μ and ν* is:

$$\mu_{adj}(\mu,\nu) = t\,[\mu_{int}(\mu,\nu),\ \sup_{x\in S}\sup_{y\in S}\ t\,[\mu(x),\nu(y),\ n_{xy}]]. \tag{2}$$

Distance between fuzzy sets uses several mathematical definitions in the crisp case. The construction principle can be applied easily, e.g. the case for distances having a direct expression in terms of mathematical morphology (e.g. nearest point distance, Hausdorff distance, or distance from a point to a set). For instance, the fuzzy equivalent of the Hausdorff distance (denoted by d_H) is

$$d_H(X,Y) = \inf\{n,\ X \subset D^n(Y) \text{ and } Y \subset D^n(X)\}. \tag{3}$$

In this formula X and Y denote two crisp sets of the considered space S, and $D^n(X)$ the dilation of size n of X. While for the nearest point distance separability and triangular inequality are not satisfied, the Hausdorff distance

is a true distance. From (3), a distance distribution (the degree to which the distance between two fuzzy sets μ and ν is less than n) can be defined by fuzzy dilation:

$$\Delta_H(\mu,\nu(n) = t[\inf_{x\in S} T[D^n(\mu)(x), c(\nu(X))], \inf_{x\in S} T[D^n(\nu)(X), c(\mu(x))]], \tag{4}$$

where c is a complementation, t a t-norm and T the dual t-conorm.

4 Application to Brain Imaging

In this section we illustrate the method for brain magnetic resonance (MR) imaging. First, we use the seven fuzzy structures shown in Figure 3. These structures may be recognized using an anatomical atlas [7], by comparing relationships between atlas structures and relationships between image structures.

The *adjacency degrees* between some of the obtained fuzzy objects in Figure 3 are given in Table II.

TABLE II. Results obtained for fuzzy adjacency, for structures given in Figure 3.

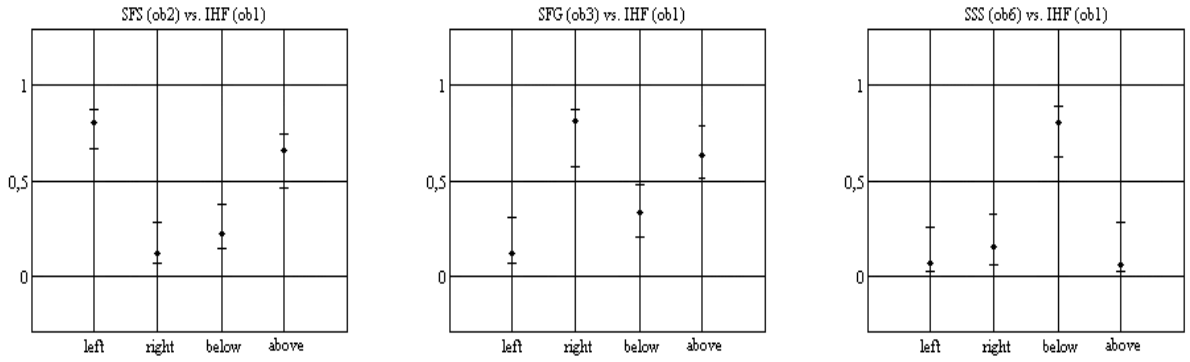
Fuzzy object 1	Fuzzy object 2	Degree of adjacency	Adjacency in the model (crisp)
1	2	0.117	0
1	6	0.463	1
1	7	0.675	1
1	5	0.035	0
1	3	0.097	0
1	4	0.026	0
3	4	0.034	0
6	7	0.087	0

High degrees were found between structures where adjacency is expected, while very low degrees are obtained in the opposite case. The results are in agreement with our expectation from the model (crisp adjacency between atlas objects), but in this case crisp adjacency would provide different results in the model and in the image. Fuzzy adjacency can indeed be used for pattern recognition purposes, of course combined with other spatial relationships.

The results are in agreement with the expectation: the model of IHF (1) provided by atlas is near from SSS (6) and POS (7), quite far from SFS (2), SFG (3) and CS (4) and very far from SPL (5). So, adjacency can be used both for identifying an object using distance as a dissimilarity measure, and for describing the spatial arrangement of objects.

The *relative position degrees* between some of the obtained fuzzy objects in Figure 3 are given in Figure 5. The three given values correspond to necessity (lowest value of the bar) and possibility (highest value of the bar) degrees, and to the average value (diamond). In (a), object SFS is mainly to the left of IHF (1) and with a significant degree to its above. Similarly, object SFG (3) (Figure b) is to the right and above of IHF, and object SSS (6) is below IHF with no ambiguity (Figure c).

Although the relative positions of objects within the brain are often constant (especially in "normal" brains) and these relationships can be expressed in terms of labels such as "left of", "below", "posterior", etc, these positions can be radically different when abnormalities are present. What tends to remain constant within the brain even in the presence of gross displacements are the relative adjacencies of structures. This observation led us to the decision *to code adjacency between structures rather than just the relative position between structures*[4][5]. Yet, since the relative position can often be specified, the adjacency links are named in accordance with the expected relative positions. This means that hypothesis about particular objects extracted from the image can be made on the basis of their expected relative positions coded in the model and then verified using more complex criteria.



a) b) c)

Figure 5. Relative positions obtained for some of the objects of Figure 3. The three values on each bar correspond to necessity (lowest value of the bar), possibility (highest value), and to the average value (diamond).

The following *geometric features* were computed for the two test images: the *perimeter P*, the *area A*, the *minimum and maximum polar distances* relative to the mean distance, R_{min}, R_{max}, the *compactness R* and the *center of gravity co-ordinates*, $\bar{x}$, $\bar{y}$, using chain codes from objects contours and appropiate formulae.

The results obtained for the geometric features are shown in TABLE III, for the objects in Figure 3.

TABLE III. GEOMETRIC FEATURES OBTAINED FOR THE FUZZY OBJECTS IN FIG. 3

Object	P	A	$\bar{x}$	$\bar{y}$	R	R_{min}	R_{max}
IHF	511	2625	119	117	7	0.040	1.998
SFS	231	1152	77	50	3	0.113	1.815
SFG	212	639	144	49	5	0.158	1.578
CS	183	1177	156	107	2	0.117	1.344
SPL	131	580	34	163	2	0.172	1.464
POS	104	297	102	166	2	0.250	1.627
SSS	79	150	122	221	3	0.312	2.030

The *relative positions* between some of the obtained fuzzy objects in Figure 3 are computed based on their centers of gravity co-ordinates by using the following rules written using the CLIPS environment. We have also computed the *relative size* and *shape* of the fuzzy object.

```
(defrule is-left
(struct ?ob1 ? ? ?x1 ? ? ? ?)
(struct ?ob2 ? ? ?x2&:(< ?x1 ?x2) ? ? ? ?) =>
(assert (struct ?ob1 is left_to struct ?ob2))
(printout t ?ob1 " is left_to struct " ?ob2 crlf)
)
(defrule is-right
(struct ?ob1 ? ? ?x1 ? ? ? ?)
(struct ?ob2 ? ? ?x2&:(> ?x1 ?x2) ? ? ? ?) =>
(assert (struct ?ob1 is right_to struct ?ob2))
(printout t ?ob1 " is right_to struct " ?ob2 crlf)
)
(defrule is-bellow
(struct ?ob1 ? ? ? ?y1 ? ? ?)
(struct ?ob2 ? ? ? ?y2&:(> ?y1 ?y2) ? ? ?) =>
(assert (struct ?ob1 is bellow_to struct ?ob2))
(printout t ?ob1 " is bellow_to struct " ?ob2 crlf)
)
(defrule is-above
(struct ?ob1 ? ? ? ?y1 ? ? ?)
(struct ?ob2 ? ? ? ?y2&:(< ?y1 ?y2) ? ? ?) =>
(assert (struct ?ob1 is above_to struct ?ob2))
(printout t ?ob1 " is above_to struct " ?ob2 crlf)
)
(defrule dimensions
(struct ?ob1 ? ?x ? ? ? ? ?)
(struct ?ob2 ? ?y&:(not(= ?x ?y)) ? ? ? ? ?) =>
(if(> ?y ?x)
then(assert (struct ?ob1 is smaller than struct ?ob2))
(printout t ?ob1 " is smaller than struct " ?ob2 crlf)
else (assert (struct ?ob1 is smaller than struct ?ob2))
(printout t ?ob1 " is bigger than struct " ?ob2 crlf))
)
(defrule complexity
```

```
(struct ?ob1 ? ? ? ? ?x ? ?) =>
(if(< ?x 2)
then(assert (struct ?ob1 has small_complexity))
(printout t ?ob1 " has small_complexity " crlf)
else (if (and (>= ?x 2) (< ?x 5))
then(assert (struct ?ob1 has medium_complexity ))
(printout t ?ob1 " has medium_complexity " crlf)
else
(assert (struct ?ob1 has higher_complexity ))
(printout t ?ob1 " ob1 has higher_complexity " crlf) ))
)
(defrule shape
(struct ?ob1 ? ? ? ? ? ?x ?y) =>
(if (< (- ?y ?x) 1.5)
then(assert (struct ?ob1 has a_regular_shape))
(printout t ?ob1 " has a_regular_shape " crlf)
else (assert (struct ?ob1 has a_unregular_shape ))
(printout t ?ob1 " has a_unregular_shape " crlf))
)
```

The *adjacency relationships* between a fuzzy object and the others can be determined by using the following rules written in CLIPS:

```
(defrule read-input
(initial-fact) =>
(printout t "Struct = ")
(assert (struct (read)))
)
(defrule find-adiacency
(struct ?x)
(struct ?x is ?dim struct ?y) =>
(printout t " struct " ?x " is " ?dim " struct " ?y crlf)
)
(defrule find-dimensions
(struct ?x)
(struct ?x is ?dim struct ?y) =>
(printout t " struct " ?x " is " ?dim " struct " ?y crlf)
)
(defrule find-complexity-shape
(struct ?x)
(struct ?x is ?com) =>
(printout t " struct " ?x " is " ?com crlf)
)
```

The *facts list* for the fuzzy objects in Figure 3 has the following form:

```
(deffacts initial-state
(struct IHF 511 2625 119 117 7 0.040 1.998)
(struct SFS 231 1152 77 50 3 0.113 1.815)
(struct SFG 212 639 144 49 5 0.158 1.578)
(struct CS 183 1177 156 107 2 0.117 1.344)
(struct SPL 131 580 34 163 2 0.172 1.464)
(struct POS 104 297 102 166 2 0.250 1.627)
(struct SSS 79 150 122 221 3 0.312 2.030)
)
```

5 Summary and Conclusions

We have shown how fuzzy sets and domain knowledge can be used in medical image processing for low level classification, in particular in a context of image fusion, and for structural representation of images, and model-based recognition. Current work aims at further developing the recognition approach sketched in this paper, based on fuzzy relations between objects, in particular in a fuzzy graph framework.

The proposed concept has been implemented and successfully used for model-driven image analysis in the domain of MRI. Promising results show the superiority of this knowledge-based approach over best traditional techniques [8] in terms of segmentation errors. The concept also has the potential for a data-driven approach. Measurements of features of fuzzy image structures can be represented using an approach based on semantic nets as fuzzy assertions [4]. Determination of some truth-value for the relation between fuzzy image structures can be exploited to attribute relations with fuzzy values (a relation may be true only for most-of-the-cases). Definition of a hyper-relational structure allows us to express the dependency of relations on facts (e.g. “CSL is inside bone only if there is no fracture”). Possible faults (pathologies) can be incorporated by using information from other sources (neurological studies) as fuzzy assertions too.

Acknowledgment

The authors would like to thank our colleague, Prof. dr. Ion Poiată, for the help in linguistically defining the brain structures and for results evaluation during our experiments.

References

[1] BHARATHI, B. and SARMA, V. V. S. (1985): Estimation of fuzzy memberships from histograms, *Information Sciences*, vol. **35**, pp. 43-59.

[2] BLOCH, I. and MAITRE, H. (1995): Fuzzy mathematical morphologies: a comparative study, *Pattern Recognition*, **28**(9), pp. 1341-1387.

[3] CIVANLAR, M. R. and TRUSSEL, H. J. (1986): Constructing membership functions using statistical data, *Fuzzy Sets and Systems*, Vol. **18**, pp. 1-13.

[4] COSTIN, H. (2004): *Biomedical Image Processing and Analysis via Artificial Intelligence and Information Fusion*, book chapter in: Knowledge Based Intelligent Systems for Health Care, T. Ichimura and K. Yoshida (Eds.), Advanced Knowledge Int. Publ. House, Australia, pp. 121-160.

[5] COSTIN, H. and ROTARIU, C. (2004): Image-understanding-based system for visual information representation using fuzzy logic, *Fuzzy Systems and A.I.-Reports and Letters*, Romanian Academy, Vol. 10, Nos.1-2, pp. 89-104.

[6] DUBOIS, D. et al. (Eds.) (1988): *Readings in Fuzzy Sets for Intelligent Systems*, Morgan Kaufmann Publ.

[7] ELSTER, A.D. (1988): *Cranial Magnetic Resonance Imaging*, Churchill Livingstone Inc..

[8] GONZALES, R.C. and WOODS, R.E. (2002): *Digital Image Processing*, Prentice Hall, New Jersey.

[9] HOHNE, K.H. et al. (1992): *Framework for the Generation of 3D anatomical atlases*, in R.A. Robb (Ed.): Proc. Visualization in Biomedical Computing, Chapel Hill, N.C., pp. 510-520.

[10] JENDRYSIK, F. et al. (1997): Fuzzy segmentation method applied to the extraction of kidney boundaries in medical images, *Proc. of EUFIT'97*, pp. 2360-2364.

[11] NAGAYAMA, I. and KUDAKA, M. (1998): Automatic detection of histological components in breast cancer image by using genetic neural network, *Proc. of Iizuka'98*, pp. 1011-1016.

[12] ROSENFELD, A. (1984): The fuzzy geometry of image subsets, *Pattern Recognition Letters*, **2**, 311-317.

[13] TEODORESCU, H.N. et al. (Eds.) (1999): *Fuzzy and Neuro-Fuzzy Systems in Medicine*, CRC Press.

[14] ZADEH, L. A. (1975): The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, **8**, pp. 199-249.

Hariton Costin
University of Medicine and Pharmacy, Iasi, Romania
Faculty of Medical Bioengineering
Address: 16 Universităţii St., Iaşi, Romania
E-mail: hcostin@iit.tuiasi.ro

Cristian Rotariu
Institute for Computer Science, Romanian Academy-Iaşi Branch
B-dul Carol I No. 11, 700506, Iasi, Romania
E-mail: crotariu@iit.tuiasi.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 163-169

A Service-Context Model allowing Dynamical Adaptation

Marcel Cremene, Michel Riveill, Christian Martel

Abstract: Mobile terminals and networks recent evolution increase the interest for new services. These services must be dynamically adapted because the context (user profile, localization, terminal and network resources, etc.) may change at runtime. The existent solutions use an adaptation control based on service-specific rules and strategies. Because of this, a service will not work correctly in a context that was not anticipated by the rules creator.
Our proposition consist in an adaptation platform that uses a service-context description in order to analyze if a service works correctly in a given context, to find the problems and the solutions. In order to adapt a service we reconfigure its architecture by inserting, replacing or moving components.

Keywords: Context, Service, Component, Architecture, Adaptation

1 Introduction

User needs and software systems complexity are continuously increasing but development time and costs need to be reduced. Thus, code reutilization and service adaptation are key concepts. In order to increase reutilization we have chosen a component oriented approach considering a service implemented as a component architecture.

Service dynamic adaptation is necessary because the context (user profile, physical resources, etc.) may evolve after the service creation, while the service is running. In order to have a better understanding about this problem, we present below an example based on a forum service. It is important to notice that this service was designed and built for a specific context C1 supposing users that speak the same language (English for instance) and are able to use a graphic interface (users able to see), also the terminal is a standard desktop with at least 14-inch screen, a WEB navigator pre-installed and a stable Internet connection are available.

If the forum is used in a context C2 different from C1, the service is no longer adapted. The following contexts are such examples and assume dynamic changes: the user may have difficulties to write messages in English so for the long phrases he prefer to use his native language, the user may be unable to watch the screen all the time because his view might be busy, the terminal is changed while using the service; this implies modifications concerning the screen size, the communication bandwidth, cost model, the user change his geographical or social position and his interest may depend on it.

In most of existent approaches, the forum service adaptation to unpredicted contexts requires human operator intervention in order to specify new rules like "*IF new_situation THEN action*" because otherwise the service does not know how to behave correctly in the new context. Some platforms like [10] for instance allow to add new rules at runtime. Anyway, because the human intervention is necessary, we can say that adaptation is less *autonomous*. Another characteristic of the existent approaches is the specialization on context aspects: some platforms concern the self-healing, other like [16] the user interface and other like [12] the content to bandwidth adaptation for instance. Thus, this is normal because a general adaptation solution is hard to do and under optimal.

Our approach was inspired by the AI(Articifial Intelligence) domain where three essential aspects are distinguished: *a)knowledge representation, b)search of solutions and c)learning*. So, our first problem was to represent the knowledge about the service and its context and the second to provide validation and solution search mechanisms; learning aspects remains for further development.

This paper is organized as it follows: in the section 2 we propose a service-context model as knowledge representation, in the section 3 we describe how we are searching the adaptation solution, the section 4 presents our prototype, the section 5 presents the related work and the section 6 contains conclusions and future work.

2 Knowledge representation: service and context

While searching models describing services and context elements we fond a problem: the existent languages related to component models like IDL, ADL, WSDL (Web Services Definition Language) [13] are not designed to allow composition with the context elements such as user, terminal, networks, etc. In order to overcome this limitation we introduce the *profiles*. In order to understand better the profiles function, a graphical illustration

is given in the figure 1. The figure depicts the three layers describing a service functioning in a specific context containing a user, a group of users and a terminal.

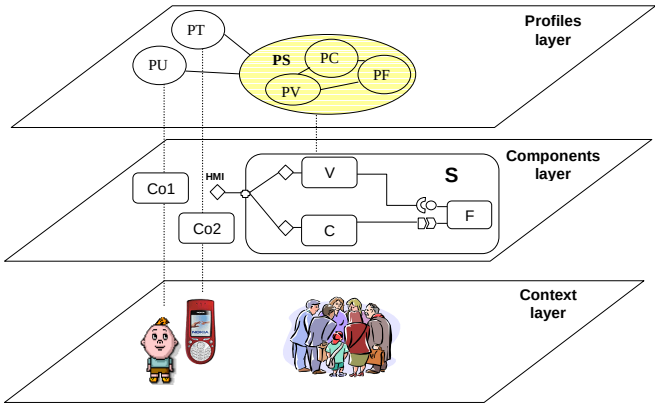


Figure 1: Service-context model - the three layers perspective

Context layer. The context layer contains elements such as: users, terminals, networks, environment (temperature, external noise or light and others). In order to facilitate the service-context assemblage, we propose to use also component models for the context elements, for instance a user is seen as a component providing HMI.

Components layer. The components layer contains the services/components described using existent models: CCM (CORBA Component Model) component model [9]. We use CORBA IDL for describing the component interfaces and FractalADL [3, 2] for architecture description. The composition is recursive like in the Fractal model, a service is also a component. In the forum case, the service S is composed by three components: C - composer, allow the user to compose new messages, V - viewer, allows the user to view the messages published on the forum and F - the forum server. The components Co1, Co2 are observer components and their role is to monitor the context elements, in this case the user profile and the terminal characteristics. A particularity of our service model is that HMI (Human Machine Interface) is also described in the same model. The service HMI is composed by the C HMI and the V HMI.

Profiles layer. The profiles layer was introduced in order to provide a unified perspective about the service and the context. In the next section we present the profile model and we explain how the profiles allow the adaptation platform to adapt the service.

2.1 Profile model

The profile model was determined starting from the interactions that exist between the involved entities: software components and context elements. We observed the same relation types as for the software component interfaces: some aspects are *offered* and others are *consumed*. We have found two types of such aspects: *resources*, for instance memory, space on the disk, space on the screen, etc; and *information* that is transmitted from one entity to another, for instance the service offers or produces messages and the user consumes them. The profile model is depicted in the figure 2. The profile elements are:

- *Component attributes* that characterize the whole component, for instance the consumed memory, the execution platform,
- *Flows* indicating an informational dependence, circulation between an input port and an output port of the component. Several flows may be related to a same component port. A flow example is the message composed at the HMI level and sent by another component port as event.
- *Flow attributes* are associated with the information offered or consumed, for example the 'language' attribute associated with a message, the 'encoding type' attribute associated with a stream, etc.

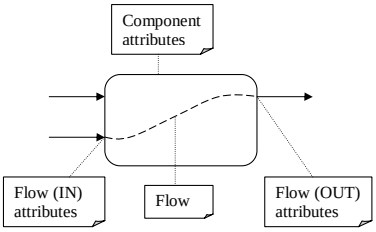


Figure 2: Profile model

The interactions existent between the components and the context elements are indicated by two aspects: a)service-independent axioms like: "the user interact with the service through the HMI", "any component consume the terminal's resources", etc; b)the attribute name is common in the component profile and context element profile, "language" for user but also for the forum service.

Any profile attribute has: a name, a definition domain, a composition operator and a comparison operator. For instance, the attribute name is "langue", its domain is discrete {FR, ED, DE, Ĕ , *, ?}, its composition attribute is ":=" (the attribution) because the value is transferred from one component to another; its comparison attribute is "=", equal, because the compatibility is verified if the same language is used.

2.2 Profile composition

The profile composition means to solve the next problem: imagine a service S composed by interconnecting N components C1ĔCN. If the component profiles PC1ĔPCN are considered as known, we want to determine the service profile PS. To determine the service profile is to determine all the profile elements: a) component (service) attributes, b) flows and c) flow attributes. The composition is done attribute by attribute and the composition operator must be known for each attribute. For instance, if C is composed by A and B; A consumes MA memory, B consumes MB memory then C also has the "memory" attribute and the composition operator is "+": MC = MA + MB.

Because the attributes are composed differently, we use different layers to compose them, like is depicted in the figure 3.

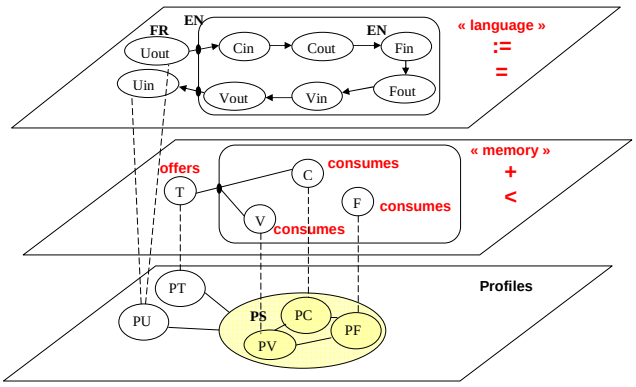


Figure 3: Profiles composition

The "memory" is an attribute associated to the whole component. The memory layer (figure 2) contains a graph that has as nodes the entities (components, context elements) having in their profiles this attribute. The arcs correspond to the relation existent between the entities from the memory point of view: components placed on a machine will consume machine memory. In our case the components C and V are installed on the client and F is installed on the server machine.

The graph has a recursive structure: a node may be expanded because an entity described by a profile may be composed by other entities.

The "language" attribute is associated with a flow. The language layer contains also a graph but in this case the nodes aren't associated with the components but rather with the component's ports. In this case the arcs indicate

the information flow. The language is composed by attribution like all the flow attributes: if the profile does not indicate a different value for an attribute, we presume that the flow keep the same values while it traverse the different components.

3 Search of solution: validation and resolution

Profiles validation

The second operator that appears in the figure 4, the "=" for the language and the "<" for the memory, are used in order to validate the profiles composition. The validation is verified for each arc in the attribute graph (see figure 3) by applying the comparison operator between the attributes values. For instance, in the "memory" layer the service consumes MS and the terminal offers MT memory, their composition is valid if MS < MT where "<" is the comparison operator.

The validation procedure is the following apply for each attribute the following operations:

- Extract the attribute layer and build the correspondent graph,
- For each arc in the graph that connects a context node with a service node verify if the attribute values are compatible by applying the comparison operator.

After the validation procedure, if the service is not adapted, a problem list is created, each problem being specified by: an attribute name and by a pair of values (that does not verify the compatibility).

Component insertion

In order to apply the insertion strategy two problems must be solved: a)Decide what component needs to be inserted in order to obtain an adapted service and b)Determine the point (or points) where the new component must be inserted in the initial service architecture.

In order to solve the problem a) we use the problem list resulted from the validation procedure and, for each problem we look for a component that may solve that problem. At this moment we know to solve problems that are related to attributes associated with the information flows: insert an additional treatment. For instance, for a problem described by the attribute "language", context value "FR" and service value "EN" we need to add an information treatment that transforms an "FR" input to an "EN" output. This is indicated in the component profile. Several solutions are possible and we may select the most suitable using a cost function for instance.

Once the solution component is found, the insertion point is determined using a search algorithm that analyzes the graph extracted for the problem related attribute, figure 4.

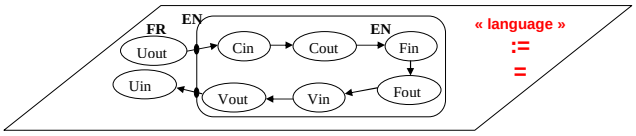


Figure 4: Adaptation validation and resolution for a "language" related problem

The algorithm tries to insert the solution component by checking the interfaces (IDL) compatibility. In this case, if it is no possible to insert the component between the first two components (the user HMI and the service HMI) the algorithm goes deeply into the service structure and tries to insert the solution component between other existent components, following the graph branches.

In the forum case, the solution component is a translator and, because its interface demands a text, it must be inserted after the composer, C, component and before the forum server, F, which imposes the language as "EN".

4 Implementation and tests

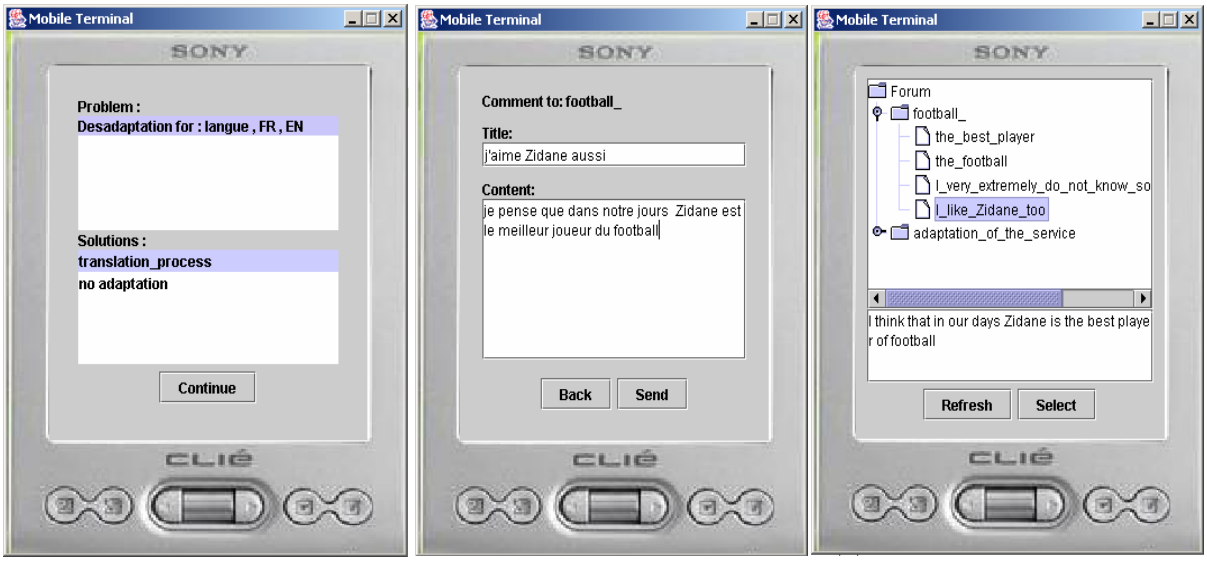
In this section we describe the prototype that we have implemented in Java in order to test our model. The CCM component model was used just for the interfaces types, for each interface type (facet, receptacle, event producer, event consumer) a Java CCM implementation was proposed. Each component has an IDL description, the HMI

is specified at the IDL level and described more detailed using a dedicated language as SunML (Simple Unified Natural Markup Language) [5], the architecture is described using FractalADL [3, 2]. The adaptation technique is based on interaction patterns described using the ISL (Interaction Specification Language) [1, 11].

The profiles are described using an XML based language. In the example depicted below we describe the profiles for the component *Translation_EN_FR* where we can identify the three elements of a profile: attributes, informational flows and flows attributes.

```
<profile component="Translation_FR_EN">
    <attribute name="memory" value="500K"/>
    <attribute name="platform"
    value="Zope 1.0"/>
    <flow id="1" portIN="trans"
    portOUT="trans">
        <attribute type="IN" name="language"
        domain="EN"/>
        <attribute type="OUT" name="language"
        domain="FR"/>
    </flow>
</profile>
```

In the figure 5 we have depicted the forum GUI evolution. After the user is authenticated, the platform detects a conflict between the user profile previously stored in a database an the service profile: the language does not verify the comparison operator that is "=".



a) **b)** **c)**

Figure 5: Forum UI

The platform proposes to user two choices: use a translator or leave the service unchanged. Supposing the user chose to use the translator from French to English, his messages are translated.

The prototype has two versions: in the first one the user language is supposed to be stored in a database, in the second the language is detected at each message (dynamic context).

5 Related work

The paper [8] describes the platform "Rainbow" that is focused on architectural adaptation. One particularity of this proposition is the use of architectural styles. The rules and the strategies are service-dependent and must be specified by an operator as we can observe in the next example:

```
// ******* Rule *******
invariant(self.responseTime < maxResponseTime)
    responseTimeStrategy(self);
// ******* Strategy ***
strategy responseTimeStrategy(ClientT C){
    let G = findConnectedServerGroup(C);
    if(query("load", G) > maxServerLoad){G.addServer(); return true;}}
```

We have the rule that tests the response time and the strategy that represent the action: add new server.

In the paper [4], A.T.S. Chan et S-N. Chuang from the University of Hong-Kong, propose the "MobiPADS" platform. This platform is based on a middleware specialized for the mobile context. A particularity of this platform is the use of complex events that combines context signals with rules and strategies. The rules are service and context dependent. The strategies are implemented using the listener mechanism. A method *notifyContextEventX* is called when a specific event is fired, this method contains the strategy implementation inside.

In [15, 14] M. Roman describe the platform "GaiaOS" for service development in active spaces. An active space is a physical space rich in terminals, sensors and other I/O devices. The rules and the strategies are also service and context dependent.

In the paper [6, 7] K. Fujii and T. Suda from the University of California propose a semantic component model called "CoSMoS" and a service generation platform called "SeGSeC". A service is assembled starting from a user phrase expressed in a natural language. Semantic graph and ontologies are used. The components models include concepts and the service assemblage follows the relations existent between these concepts. This proposition does not take yet into account a dynamic context.

6 Conclusions and perspectives

In this paper we have proposed an AI approach for service dynamically reconfiguration at architectural level (by adding, replacing or moving components). The main contribution of our work is the unified service-context model based on profiles, the composition and validation mechanism. The system may automatically discover different adaptation choices.

All components and context elements must have profiles and these profiles must be specified by a human operator. The profiles formalization and their composition algebra is under development.

In perspective we intend to solve the following problems: strategy selection, search algorithms improvement, validate the model with more adaptation examples, develop the profiles semantic, use AI tools such as inference engines, reinforcement based learning.

References

[1] M. Blay-Fornarino, D. Ensellem, A. Occello, A.-M. Pinna-Dery, M. Riveill, J. Fierstone, O. Nano, and G. Chabert. Un service d'interactions : principes et implémentation. In *Journee Composants*, INRIA, Grenoble, France, October 2002.

[2] E. Bruneton. Fractal adl tutorial, version 1.2. Technical report, France Telecom RD, france, March 2004.

[3] E. Bruneton, T. Coupaye, and J. B. Stefani. Recursive and dynamic software composition with sharing. In *ECOOP Workshop on Component-Oriented Programming*, pages ??–??, Malaga, Spain, 2002.

[4] A. T. S. Chan and S. N. Chuang. Mobipads: A reflective middleware for context-aware mobile computing. *IEEE Trans. Software Eng.*, 29(12):1072–1085, 2003.

[5] J. Fierstone, A.-M. Dery-Pinna, and M. Riveill. Architecture logicielle pour l'adaptation et la composition d'imh. mise en oeuvre avec le langage sunml. Technical report, lab. I3S, ESSI, Université de Nice, Sophia Antipolis, France, January 2003.

[6] K. Fujii and T. Suda. Component service model with semantics (cosmos): A new component model for dynamic service composition. In *International Symposium on Applications and the Internet Workshops (SAINTW'04)*, pages 348–355, Tokyo, Japan, 2004.

[7] K. Fujii and T. Suda. Dynamic service composition using semantic information. In *2nd International Conference on Service Oriented Computing (ICSOC 04)*, pages ??–??, New York City, NY, USA, 2004.

[8] D. Garlan, S.-W. Cheng, A.-C. Huang, B. R. Schmerl, and P. Steenkiste. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *IEEE Computer*, 37(10):46–54, 2004.

[9] O. M. Group. Corba components : Joint revised submission. Technical report, Sun Microsystems Inc. 2550 Garcia Avenue, Mountain View, CA 94043, http://java.sun.com/beans, August 1999.

[10] K. John and C. Vinny. Chisel: A policy-driven, context-aware, dynamic adaptation framework. In *Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 3–13, Lake Como, Italy, June 2003.

[11] M. R. Mireille Blay-Fornarino. Un service d'interactions. *Technique et science informatiques, 0752-4072, Revue des sciences et technologies de l'information*, 23(2), 2004.

[12] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Agile application-aware adaptation for mobility. In *Sixteen ACM Symposium on Operating Systems Principles*, pages 276–287, Saint Malo, France, 1997.

[13] W. Note. Web services description language (wsdl) 1.1. Technical report, The ObjectWeb Consortium, Mars 2001.

[14] M. Roman. *An Application Framework for Active Space Applications*. PhD thesis, University of Illinois at Urbana-Champaign, 2003.

[15] M. Roman, C. K. Hess, R. Cerqueira, K. Narhstedt, and R. H. Campbell. Gaia: A middleware infrastructure to enable active spaces, uiucdcs-r-2002-2265 uilu-eng-2002-1709. Technical report, University of Illinois at Urbana-Champaign, USA, February 2002.

[16] S. Trewin, G. Zimmermann, and G. Vanderheiden. Abstract user interface representations: how well do they support universal access? In *Proceedings of the 2003 conference on Universal usability*, pages 77–84. ACM Press, 2003.

Marcel Cremene
Technical University of Cluj-Napoca
Department of Communication
Address: 26, Baritiu, Cluj-Napoca, Romania
E-mail: cremene@com.utcluj.ro

Michel Riveill
University of Nice, ESSI
Address: Sophia Antipolis, France
E-mail: riveill@unice.fr

Christian Martel
University of Savoie, Lab. SysCom
Address: Chambery, France
E-mail: cmart@univ-savoie.fr

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 170-174

Incremental Horizontal Fragmentation: A new Approach in the Design of Distributed Object Oriented Databases

Adrian Sergiu Darabant, Alina Câmpan, Horea Todoran, Gabriela Şerban

Abstract: Distributed relational or more recently object-oriented databases usually employ data fragmentation techniques during the design phase in order to split and allocate the database entities across the nodes of the system. Most of the design algorithms are usually static and do not take into account the system evolution: data updates and addition of new applications. As this is an important issue in modern databases, we propose in this paper an incremental method for dynamically re-fragmenting an existing database as it evolves over time. We prove that by incrementally changing the database schema and allocation over time we improve database availability. This is a direct result of the reduced amount of time needed for incremental fragmentation as compared to a full fragmentation.

Keywords: distributed databases, incremental fragmentation, clustering

1 Introduction

We present this in this paper a new method for incremental horizontal fragmentation in distributed object oriented databases. The fragmentation process is an important step in distributed database design and greatly influences the database overall performance factor. A poorly fragmented database will usually perform worse than a centralized database because of the reduced parallelism degree and increased data transport costs. We have already presented in [1, 2, 3, 4] a novel approach in horizontal object-oriented database fragmentation. The proposed methods deal with all the complex aspects of the Object Oriented (OO) data model, but can be successfully applied to a relational database as well. The proposed fragmentation methods all center around relieving the database administrator from doing statistical estimations about the database: significance of each application, precedence of the nodes of the system, potential size of data on each node, etc. The initial fragmentation phase is not sufficient however for dynamic databases that evolve over time. The state and values of the existing entities change over time. The application set that accesses the database evolves over time as well. New queries enter the system and the existing ones might evolve. These changes invalidate in time the original distributed schema of the database. For obtaining the fragmentation that fits the new user applications set, the original fragmentation scheme can be applied from scratch, an undesirable alternative from the point of view of processing effort, extended database maintenance and unavailability time. To our knowledge, there are no practical approaches for incrementally maintaining an efficient database fragmentation. We propose in this paper an incremental technique to cope with the evolving user application set. Namely, we handle here the case when new user applications arrive in the system and the current primary fragments must be accordingly adapted.

2 Quantification of the object model of the database

In this section we will shortly review the vector space model used to quantify the database properties. An extended presentation of the vector model can be found in [1, 2, 3, 4].

In our model classes are organized in an *inheritance hierarchy*, in which a subclass is a specialization of its superclass. A class C is an ordered tuple $C = (K, A, M, I)$, where A is the set of object attributes, M is the set of methods, K is the class identifier and I is the set of instances of class C.An object O is an *instance* of a class C if C is the most specialized class associated with O in the inheritance hierarchy. An object O is *member* of a class C if O is instance of C or of one of subclasses of C. An entry point into a database is a meta-class instance bound to a known variable in the system. An entry point allows navigation from it to all classes and class instances of its sub-tree (including itself). There are usually more entry points in an *OODB*.

Given a complex hierarchy H, a path expression P is $C_1.A_1 \ldots A_n$, $n \geq 1$ where: C_1 is an entry point in H, A_1 is an attribute of class C_1, A_i is an attribute of class C_i in H such that C_i is the domain of attribute A_{i-1} of class C_{i-1} ($1 \leq i \leq n$).

The fragmentation and allocation of an object oriented database aim to optimize the execution of a set of user queries/applications. In general a *query* is a tuple with the following structure: *q=(Target class, Qualification*

clause), where: *Target class* specifies the class over which the query returns its object instances; *Qualification clause* is a logical expression over the class attributes in conjunctive normal form. The logical expression is constructed using simple predicates: $attribute \Theta value$ where $q \in \{<, >, \leq, \geq, =, \neq\}$.

Let $Q = \{q_1, \ldots, q_t\}$ be a set of queries in respect to which the fragmentation has to be performed. Let $Pred_Q = \{p_1, \ldots, p_q\}$ be the set of all simple predicates Q is defined on. Let $Pred_Q(C) = \{p \in Pred_Q |$ p imposes a condition to an attribute of class $C\}$ be the set of predicates that apply to class C. Given two classes C and C', where C' is subclass of C, $Pred_Q(C') \supseteq Pred_Q(C)$. The reasons for this condition inheritance are explained in [1, 2, 3].

To each object O_i in the set $Inst(C)$ of all instances of class C, $i = 1..m$, $m = |Inst(C)|$, we associate an *object-condition vector* $a_i = (a_{i1}, \ldots, a_{is})$, where $Pred_Q(C) = \{p_1, \ldots, p_s\}$:

$$a_{ij} = \begin{cases} 0, \ if \ p_j(O_i) = false \\ 1, \ if \ p_j(O_i) = true \end{cases}$$

In our method objects will be grouped together in fragments using clustering methods so that objects within a fragment have high similarity with each other and low similarity with objects in other groups. Similarity and dissimilarity between objects are calculated using metric or semi-metric functions, applied to the *object-condition vectors* that characterize objects. We use in this paper the *Euclidian distance* for measuring objects similarity:

$$d_E(a_i, a_j) = \sqrt{\sum_{l=1}^{s} (a_{il} - a_{jl})^2},$$

where a_i, a_j are the *object-condition vectors* of O_i, $O_j \in Inst(C)$.

3 Incremental clustering based fragmentation using CBIC

3.1 Initial fragmentation phase

When passing from a centralized database to a distributed one, an initial fragmentation is required. In our approach [1, 2, 3, 4], given a set $Q_{init} = \{q_1, \ldots, q_p\}$ of queries, the initial fragmentation phase of the object set $Inst(C)$ of class C requires first that objects in $Inst(C)$ to be modelled as described above. Then a clustering method (*k-means*) or *hierarchical clustering* is applied over the vector space describing $Inst(C)$, and the resulting clusters represent the fragments for class C. Afterwards as the database evolves over time, when certain conditions are met a new fragmentation might be required. This will happen often for highly dynamic databases and less often for mostly static data.

3.2 Incremental Fragmentation Based on Applications Change

The existing fragmentation of the distributed object oriented database was developed to optimize the execution of the initial query set, Q_{init}. When new queries arrive into system $Q_{new} = Q_{init} \cup \{q_{p+1}, \ldots, q_t\}, t > p$, the current fragmentation must be adapted. We apply in this case an incremental, *k-means* based clustering method, *Core Based Incremental Clustering* (*CBIC*) [5, 6].

The extension of the query set Q_{init} to Q_{new} means that for a number of classes in the database, their associated set of predicates increases. These classes have to be re-fragmented to fit the new query set. Let C be such a class, for which $Pred_{Q\_init}(C) = \{p_1, \ldots, p_n\}$ evolves to $Pred_{Q\_new}(C) = Pred_{Q\_init}(C) \cup \{p_{n+1}, \ldots, p_s\}$. Consequently, the *object-condition vector* for each object $O_i \in Inst(C)$ is extended as follows:

$$a_i' = (\underbrace{a_{i1}, \ldots, a_{in}}_{initial\ object-condition\ a_i\ of\ O_i}, a_{i,n+1}, \ldots, a_{is})$$

The *CBIC* method starts from the existing partitioning of $Inst(C)$ into clusters (the existing fragments established by applying *k-means*). Let $\{K_1, K_2, \ldots, K_p\}$ be the initial fragments of $Inst(C)$, $K_i \cap K_j = \emptyset, i \neq j, \bigcup_{l=1}^{p} K_l = Inst(C)$. *CBIC* determines then $\{K_1', K_2', \ldots, K_p'\}$ the new partitioning of objects in $Inst(C)$ after query set extension. It starts from the idea that, when adding few components (features, attributes) to the object-condition vectors such that these components don't bring to much information in the system, then the old arrangement into clusters is close to the new one. The algorithm determines then those objects within each fragment K_i that have a considerable chance to remain together in the same cluster. They are those objects that, after feature extension, still remain closer to the centroid (cluster mean) of cluster K_i. These objects form what is called the core of cluster K_i, denoted by $Core_i$. Note: the centroid of K_i to which we report the object after extension is calculated as the mean of the extended object-condition vectors of objects in K_i.

The cores of all fragments K_i, $i = 1..p$, will be the new initial clusters from which the iterative partitioning process begins. Next, *CBIC* proceeds in the same manner as the classical *k-means* does. The *CBIC* algorithm is presented below and variants can be found in [5, 6].

```
Algorithm Core Based Adaptive k-means is
Input: - the set X = {O_1,...,O_n} of m-dimensional previously clustered
       objects,
       - the set X' = {O'_1,...,O'_n} of (m+s)-dimensional extended objects
       to be clustered; O'_i has the same first m components as O_i,
       - the metric d_E between objects in a multi-dimensional space,
       - p, the number of desired clusters,
       - K = {K_1,...,K_p} the previous partition of objects in X,
       - noMaxIter the maximum number of iterations allowed.
Output: - the new partition K' = {K'_1,...,K'_p} for the objects in X'.
Begin
   For all clusters K_j ∈ K
      Calculate Core_j = (StrongCore_j ≠ ∅)?StrongCore_j : WeakCore_j
      K'_j = Core_j
      Calculate f'_j as the mean of objects in K'_j
   EndFor
   While (K' changes between two consecutive steps) and
         (there were not performed noMaxIter iterations) do
      For all clusters K'_j do
         K'_j = {O'_i | d(O'_i, f'_j) ≤ d(O'_i, f'_r), ∀r, 1 ≤ r ≤ p, 1 ≤ i ≤ n}
      EndFor
      For all clusters K'_j do
         f'_j = the mean of objects in K'_j
      EndFor
   EndWhile
End.
```

4 Results and Experiments

As experiments we conducted a number of fragmentations for different sized databases with different schemas. The smallest considered one has five classes each with 30 to 130 instances. The largest configuration is for a schema with the same five classes with an average of 40,000 instances and the largest instance number for a class is around 128,000. We only considered schemas with an unrealistic number of classes: 5 to 10 because the fragmentation algorithms are not influenced by the number of classes: each class is separately fragmented. Dependencies between classes induced by inter-class relationships are represented in the vector space model and have only a linear influence on the results of the algorithm. That means that multiplying the number of classes will increase in a linear manner the number of iterations for the algorithms (both the fully-fledged one as the incremental one). In all experiments we started with a number of running applications and doubled their number in time. The numbers bellow are for 12 starting applications with 20 running applications at the end.

4.1 Number of Iterations

In this section we shortly present the number of iterations required by both the full and the incremental algorithms to produce the final database fragments for each class. As experiments show, the result is generally reached by *CBIC* more efficiently than running *k-means* again from the scratch on the feature-extended object set.

It can be seen that the incremental CBAk only needs 70% of the number of iterations of the k-Means algorithm.In the following section we will compare the resulting fragmentation quality when applying the full k-Means method and the incremental algorithm. We should note here that the time required to run a full algorithm iteration is directly proportional with the number of class instances. So the incremental fragmentation will probably not pay its costs in the case of small sized databases.

Table 1: Comparative results for the *CBAk* and *k-means* algorithms

Experiment	UnderGrad	Grad	Prof	Staff	Researcher
No objects	128000	26000	30000	8000	6000
No attributes (m+s)	8	6	7	3	4
No of new attributes (s)	3	2	3	1	1
No iterations k-means for (m+s) attributes	30	20	20	20	10
No iterations CBAk for (m+s) attributes	20	20	10	10	10

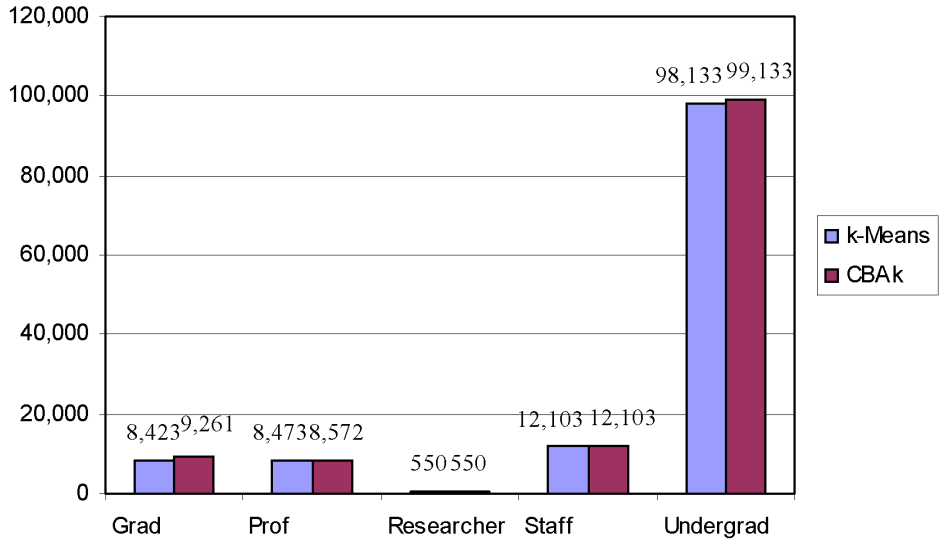


Figure 1: The fragmentation costs for the CBAk(incremental) and k-Means(full) fragmentation methods

4.2 Fragmentation Quality

In order to compare the fragmentation quality we use a cost function applied to the resulting fragments. The cost function is the one presented in [1, 3, 4] and is derived from the Partition Evaluator as proposed by Chakravarthy in [7]. The cost function evaluates the cost of accessing data by the existing applications when the fragments are each allocated to the node where they are most used. We considered for this a distributed system with 4 nodes. The cost function as introduced in [1] is presented below:

$PE(C) = EM^2 + ER^2$, **(Total cost)** where:

$EM^2(C) = \sum_{i=1}^{M} \sum_{t=1}^{T} freq_{ts}^2 \times |Acc_{it}| \times (1 - \frac{|Acc_{it}|}{|F_i|})$ **(Local processing cost)**

$ER^2(C) = \sum_{t=1}^{T} \{ \sum_{s=1}^{S} \sum_{i=1}^{M} freq_{ts}^2 \times |Acc_{it}| \times \frac{|Acc_{it}|}{|F_i|} \}$ **(Remote accessing cost)**

As presented in [1] the EM term computes the costs induced by accessing data located on the same node as the running application. The ER term computes the costs induced by accessing data located on other nodes than the node on which the application is running. As the value of the cost is smaller the fragmentation quality is better. The average obtained results are represented in the figure 1. The figure represents the fragmentation costs for a full re-fragmentation using the k-Means algorithm when the number of applications increases from 12 to 20 and the costs induced by the incremental fragmentation of the existing database in the same conditions. It can be seen that the costs induced by the incremental fragmentation are slightly larger or the same for each class. However the differences are very small. Compared to the fact that the number of iterations is reduced by around 30% we conclude that generally the obtained results are better than in the case of full database refragmentation. We support

our affirmations by considering that maintaining the database availability over time is more important than gaining a very small performance but having a much greater general database downtime.

5 Summary and Conclusions

We proposed in this paper a study on the applicability of two (incremental and full) fragmentation methods in the dynamic design of a distributed object oriented database. As our experiments show the incremental method can be effective and efficient, by reducing the maintenance effort and reducing the time for database tuning. Even if the CBIC method proves to be efficient in most of the cases, there are situations when is more appropriate to apply a full fragmentation. This is desirable when the number of applications accessing the data changes significantly in given amount of time without intervening incremental fragmentations. This types of context evolutions do not scale well with the incremental fragmentation method as the information gain brought by the new applications is too important. Generally, in this cases we cannot observe an improvement in the number of the algorithm steps, while still maintaining worse fragmentation quality than in the full re-fragmentation case.

References

[1] Darabant, A.S., Campan, A., Semi-supervised learning techniques: k-means clustering in OODB Fragmentation, IEEE International Conference on Computational Cybernetics ICCC 2004, Vienna University of Technology, Austria, August 30 - September 1, 2004, pp. 333–338.

[2] Darabant, A.S., Campan, A., Hierarchical AI Clustering for Horizontal Object Fragmentation, In Proc of Int. Conf. of Computers and Communications, Oradea, May, 2004, pp. 117–122.

[3] Darabant, A.S., Campan, A., AI Clustering Techniques: a New Approach to Object Oriented Database Fragmentation, in Proceedings of the 8th IEEE International Conference on Intelligent Engineering Systems, Cluj Napoca, 2004, pp. 73–78.

[4] Darabant, A.S., Campan, A., Cret, O., Hierarchical Clustering in Object Oriented Data Models with Complex Class Relationships, in Proceedings of the 8th IEEE International Conference on Intelligent Engineering Systems, Cluj Napoca, 2004, pp. 307–312.

[5] Serban, G., Campan, A., Core Based Incremental Clustering, Studia Universitatis “Babeş-Bolyai”, Informatica, XLXI(2), 2005, pp. 89–96.

[6] Serban, G., Campan, A., Incremental Clustering Using a Core-Based Approach, in Proc. of the 20th International Symposium on Computer and Information Sciences (ISCIS’05), Istanbul, Turkey, 2005 (to appear).

[7] Chakravarthy, S., Muthuraj, J., Varadarajan, R., Navathe S.B - An Objective Function for Vertically Partitioning Relations in Distributed Databases and its Analysis, In Distributed and Parallel Databases, 2(1) pp 183-207, 1993.

Adrian Sergiu Darabant, Alina Câmpan, Gabriela Şerban, Horea Todoran
Babeş Bolyai University - Cluj Napoca
Department of Computer Science
Address: Kogalniceanu 1, Cluj Napoca, 400090
E-mail: {dadi,alina,gabis}@cs.ubbcluj.ro, htodoran@euro.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 175-180

Alternative Algorithms for Finding the Conex Components for a Graph

Adrian Deaconu

Abstract: Two algorithms for finding the conex components are presented. Sequentially, the nodes and the arcs of the initial graph are transformed. At the end of the algorithm, the graph has each node a set of nodes N_x of the initial graph, where x is the representative node for N_x. The final graph has no arcs. Each set of nodes N_x is a conex component of the initial graph.

1 Introduction

Let $G = (N, A)$ be a digraph with n nodes and m arcs.

The main idea for the algorithms presented in this paper for finding the conex components is that the number of nodes and the arcs of the initial graph G are modified. At the end of the algorithm the graph has as nodes a set of nodes of the initial graph and no arcs. Each set of nodes is a conex component of the initial graph.

2 The first algorithm - elimination of arcs

The algorithm starts with a graph $G' = \{N', A'\}$ that has $N_x = \{x\}$ as nodes, $\forall x \in N$ and (N_x, Ny) as arcs, $\forall (x,y) \in A$, i.e., $N' = \{N_x | x \in N\}$ and $A' = \{(N_x, N_y) | x, y \in N, (x,y) \in A\}$.

The algorithm starts also with the set U containing all the nodes of the graph G, i.e., $U = N$. At each iteration of the algorithm a node x is randomly chosen from the set U. The algorithm ends when the set U becomes empty.

If there is a an arc $(N_x, N_y) \in A', x \neq y$, the arc (N_x, N_y) is eliminated from A'. Every arc $(N_z, N_y) \in A'$ is eliminated and, instead, the arc (N_z, N_x) is introduced in A', because the nodes from N_y will be introduced into the set N_x. Every arc $(N_y, N_z) \in A'$ are also eliminated from the graph G' and, instead of it, the arc (N_x, N_z) are introduced in A'. At the end of iteration, the nodes from N_y are added into the set N_x, N_y becomes empty and the node y leaves the set U.

If there is no more arcs $(N_x, N_y) \in A', x \neq y$, then the node x leaves the set U and there is no arc $(x,y) \in A$, with $y \in N - N_x$. So, the set N_x contains the nodes of a conex component of G.

The algorithm ends when the set U becomes empty and, so, all the conex components are found.

The pseudo-code of the algorithm is:

```
For k := 1 to n do
    N_k := {k};
End for;
U := N;
B := A;
While U ≠ Φ do
    Select a node x from the set U;
    If ∃(x,y) ∈ B and x ≠ y then
        B := B − {(x,y)};
        For each (y,z) ∈ B do
            B := B − {(y,z)};
            B := B ∪ {(x,z)};
        End for;
        for each (z,y) ∈ B do
            B := B − {(z,y)}
            B := B ∪ {(z,x)}
        End for;
        N_x := N_x ∪ N_y;
        Ny := Φ;
        If y ∈ U then
            U := U − {y};
        End if;
```

```
        else
            U := U − {x};
        End if;
    End while;
    For k := 1 to n do
        If N_k ≠ Φ then
            N_k is a conex component;
        End if;
    End for;
```

In order not lose the set A, a set B is considered, which is equal to A at the beginning of the algorithm.

The correctness of the algorithm

It is easy to see that the execution of the algorithm ends in a finite number of elementary steps, because every arc of the graph G' is considered at most once and at every iteration of the algorithm at least one arc of A' is eliminated. When there is no more arc with one extremity in N_x, the node x leaves the set U. The algorithm ends when U is empty.

We are going to prove now that at the end of the algorithm every set N_k which is not empty conteins a conex component of the graph. It is necessarily and sufficient to prove that for each set N_k:

1. There is a path in G from the node u to the node v, for each $u, v \in N_k$.
2. There is no path in G from a node $u \in N_k$ to a node $v \in N - N_k$.

We consider that at the iteration p of the algorithm the first affirmation is true. At the iteration $p+1$ we have two situations:

i. The nodes $x, y \notin N_k$. In this situation the set N_k does not change and the affirmation remains true.
ii. The node $x \in N_k$. This implies that $N_k = N_x$.

Let u, v be in $N_x \cup N_y$.
If $u, v \in N_x$ or $u, v \in N_y$, then there is a path from u to v (the affirmation is true in the previous iterations of the algorithm).
If $u \in N_x$ and $v \in N_y$, then $u = x$ and $v = y$. So, there is a path in G from u to v.

For the second affirmation, we suppose that at the end of the execution of the algorithm there is a set N_k so that there is $u \in N_k$ and $v \in N - N_k$ and there is a path from u to v in G. Let $L = (u = n_1, n_2, ..., n_s = v)$ be this path, where $n_1, n_2, ..., n_p \in N_k, 1 \leq p < n$ and $n_{p+1} \notin N_k$. This means that the arc (n_p, n_{p+1}) does not leave the set B. It results that $N_k = N_h$, because $N_i \cap N_j, \forall i, j \in N, i \neq j$ and $N = N_1 \cup N_2 \cup ... \cup N_n$. This implies that $n_{p+1} \in N_k$, but this fact contradicts the initial supposition $n_{p+1} \notin N_k$.

So, both affirmations (i. and ii.) are proven and this means that the algorithm is correct. We have the following theorem:

Theorem 1. *The algorithm finds the conex components of the graph G.*

The complexity of the algorithm

The time complexity of the algorithm is given by the following theorem:

Theorem 2. *The time complexity of the algorithm is* $O(n^2)$.

Proof: First, let's observe that the algorithm does not consider all the arcs of the graph G.
We have:
At each iteration of the algorithm a node (x or y) leaves the set U.
A node x can be selected from U in $O(1)$ time complexity.

An arc $(x,y) \in B, x \neq y$ can be found in a complexity of $O(n)$ and the decision if there is such an arc can be also done in $O(n)$ time complexity (if the nodes of G are given by the adjacency matrix).

The two for loops of the algorithm are executed in $O(n)$ time complexity (if the adjacency matrix is considered).

The nodes from N_x can be added into the set N_y in $O(1)$ time complexity (copying a memory zone to another address).

A node leaves a set N_x in $O(n)$.

So, the total time complexity of the algorithm is $O(n^2)$.

3 The second algorithm - discussion of arcs

The idea of this algorithm is the discussion of all arcs of the initial graph. The sets $N_k, k \in N$ are also constructed. Each arc (x,y) is tested if its extremities x and y belong to the same set N_k. If they are in two different sets N_k and respectively $N_h, k \neq h$, then the set with less nodes is added to the set with more nodes and the set with less nodes becomes empty. In order to identify easily the set which contains a node, a characteristic vector a is used. If $a[x] = k$, then this means that $x \in N_k$. Moreover, in order to compare fast the number of nodes in the set N_k and in the set N_h, another vector denoted l is used, i.e., $l[k]$ is the number of nodes in the set N_k.

The algorithm is:

```
For k := 1 to n do
    N_k := {k};
    l_k := 1;
    a_k := i;
End for;
For each arc (x,y) ∈ A do
    If a_x ≠ a_y then
        If la_x > la_y then
            N_{a_x} := N_{a_x} ∪ N_{a_y};
            l_{a_x} := l_{a_x} + l_{a_y};
            i := a_y;
            For k := 1 to l_{a_y} do
                a[N_{a_y}[k]] := a_x;
            End for;
        else
            N_{a_y} := N_{a_y} ∪ N_{a_x};
            l_{a_y} := l_{a_y} + l_{a_x};
            i := a_x;
            For k := 1 to l_{a_x} do
                a[N_{a_x}[k]] := a_y;
            End for;
        End if;
        l_i := 0;
    End if;
End for;
For k := 1 to n do
    If l_k > 0 then
        N_k is a conex component;
    End if;
End for;
```

The correctness of the algorithm

Two extremities of an arc belong to the same conex component. The algorithm discuss all the arcs of the graph. Each arc is verified if its extremities are in the same set N_k. If they are in the same set, then nothing is done. If they

are not in the same set, then the two sets put their nodes together. At the end of the algorithm, each not empty set $N_k, k \in N$ contains the nodes linked with a path. So, they contain the conex components of the graph.

Theorem 3. *The second algorithm (discusion of arcs) finds the conex components of the graph G.*

The complexity of the algorithm

Theorem 4. *The time complexity of the algorithm is* $O(max\{m, n \cdot log(n)\})$.

Proof: We have:

i. The test $a_x \neq a_y$ is passed at most $n-1$ times, because each time it is passed the nodes from a set are added into another set and the first set becomes empty.

ii. O(1) time complexity is needed for $N_{a_x} := N_{a_x} \cup N_{a_y}$ or $N_{a_y} := N_{a_y} \cup N_{a_x}$, because $N_{a_x} \cap N_{a_y} = \Phi$.

iii. $a[N_{a_x}[k]] := a_y$ or $a[N_{a_y}[k]] := a_x$ are executed at most $\frac{n}{2} \cdot [log_2 n]$ times, because the set with less nodes is added to the set with more nodes. For instance, for a conex graph with $n = 8 = 2^3$ nodes, the worth case is when after 4 adds there are 4 sets N_k, each having 2 nodes and 4 times is executed $a[N_{a_x}[k]] := a_y$ or $a[N_{a_y}[k]] := a_x$. After 2 more adds there are 2 sets, each having 4 elements and 4 times is executed $a[N_{a_x}[k]] := a_y$ or $a[N_{a_y}[k]] := a_x$. Finally, after the last add, 4 times is executed $a[N_{a_x}[k]] := a_y$ or $a[N_{a_y}[k]] := a_x$. So, $a[N_{a_x}[k]] := a_y$ or $a[N_{a_y}[k]] := a_x$ is executed $4+4+4 = 4 \cdot 3 = \frac{8}{2} \cdot log_2 8 = \frac{n}{2} \cdot log_2 n$ times.

iv. '`if` $a_x \neq a_y$' statements are executed m times.

So, it is easy to see now that the time complexity of the algorithm is $O(m+n+n \cdot log(n)) = O(max\{m, n \cdot log(n)\})$.
In most of the cases $m \geq n \cdot log(n)$. So, the time complexity of the algorithm can be considered $O(m)$.

4 Examples

The progress of the first algorithm (elimination of arcs) applied to a graph with 12 nodes is shown in the figures 1 and 2.

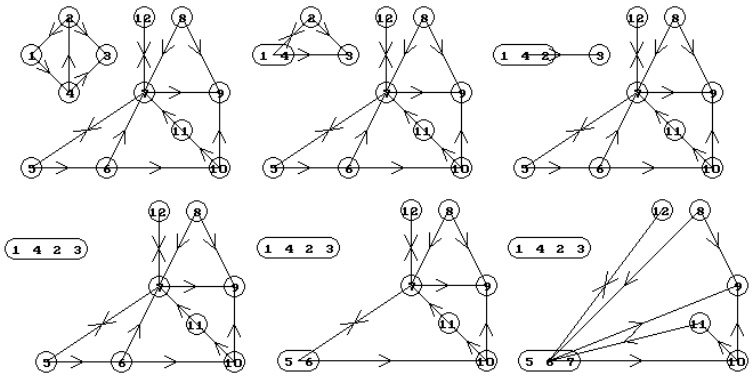


Figure 1: This is the caption for the graphic

If the second algorithm (discussion of arcs) is applied to the initial graph from the figure 1, then we have:

$a = (1,2,3,4,5,6,7,8,9,10,11,12)$
$N_1 = \{1\}; N_2 = \{2\}; N_3 = \{3\}; N4 = \{4\}; N_5 = \{5\}; N_6 = \{6\}; N_7 = \{7\}; N_8 = \{8\}; N_9 = \{9\}; N_{10} = \{10\}; N_{11} = \{11\}; N_{12} = \{12\}$.

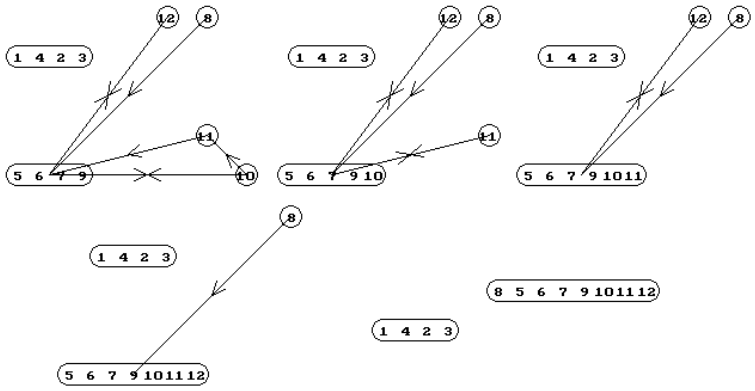


Figure 2: This is the caption for the graphic

In order to watch the progress of the algorithm, at each iteration of the algorithm it is presented the arc which is discussed and the vector a, if it is modified:

1. $(1,4) => a = (4,2,3,4,5,6,7,8,9,10,11,12)$
2. $(2,1) => a = (4,4,3,4,5,6,7,8,9,10,11,12)$
3. $(2,3) => a = (4,4,4,4,5,6,7,8,9,10,11,12)$
4. $(4,2)$
5. $(4,3)$
6. $(5,6) => a = (4,4,4,4,6,6,7,8,9,10,11,12)$
7. $(5,7) => a = (4,4,4,4,6,6,6,8,9,10,11,12)$
8. $(6,7)$
9. $(6,10) => a = (4,4,4,4,6,6,6,8,9,6,11,12)$
10. $(7,5)$
11. $(7,9) => a = (4,4,4,4,6,6,6,8,6,6,11,12)$
12. $(7,12) => a = (4,4,4,4,6,6,6,8,6,6,11,6)$
13. $(8,7) => a = (4,4,4,4,6,6,6,6,6,6,11,6)$
14. $(8,9)$
15. $(10,9)$
16. $(10,11) => a = (4,4,4,4,6,6,6,6,6,6,6,6)$
17. $(11,7)$
18. $(12,9)$

So, the conex components are:

$N_4 = \{4,1,2,3\}$ and $N_6 = \{6,5,7,10,9,12,8,11\}$.

5 Conclusion

Two algorithms for finding the conex components of a graph have been presented. If the adjacency matrix is considered, then these two algorithms have the same time complexity as the well-known algorithm using a graph search. If the adjacency lists are used, then in most of the cases $m \geq n \cdot log(n)$ and the second algorithm (discussion of arcs) has the complexity of $O(m)$.

So, our intension was not to improve the time complexity of the well-known algorithm, but to present two alternative algorithms with similar complexity.

References

[1] E. Ciurea, *Algoritmi. Introducere in algoritmica grafurilor,* Editura Tehnica, Bucuresti, 2001.

[2] T. H. Corman, C. E. Leiserson, R. L. Rivest, *Introduction to Algorithms,* MIT Press, Cambridge, Messachusetts, 1990.

[3] C. Croitoru, *Tehnici de baza in programarea combinatorie,* Editura Universitatii Al. I. Cuza, Iasi, 1992.

[4] S. Even, *Graph Algorithms,* Computer Science Press, Rockville, 1979.

Adrian Deaconu
University *Transilvania*
Theoretical Computer Science Department
Address: 50, Iuliu Maniu, Brasov, Romania
E-mail: Deaconu@info.unitbv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 181-186

On the Cyclic Subgroupoids of a Brandt Groupoid

Marian Degeratu, Gheorghe Ivan, Mihai Ivan

Abstract: We refer to the groupoids in the sense of Brandt. In this paper we give a program on computer for finding of the cyclic subgroupoids of a finite groupoid.

Keywords: groupoid, subgroupoid, cyclic subgroupoid, cyclic groupoid

1 Introduction

The concept of groupoid has introduced by H. Brandt [Math. Ann. **96**, 1926, 360 -366]. In the language of categories, a groupoid is a small category in which all morphisms are invertible. More details about groupoids can be find in the papers [1], [3], [7].

The first section is dedicated to the concept of cyclic subgroupoid. In the second section we give an algorithm for finding the cyclic subgroupoids of a groupoid. This algorithm is implemented on computer and we obtain the program *BGroidAP*6. We illustrate the utilisation of this program on some finite groupoids.

The program exposed in this paper plays an essential tool for the study of finite groupoids.

2 Cyclic groupoids

Let (G, G_0) be a pair of nonempty sets with $G_0 \subseteq G$, endowed with the surjections $\alpha, \beta : G \to G_0$, called the *source* and the *target*, a (*partial*) *composition law* $\mu : G_{(2)} \longrightarrow G, (x,y) \longrightarrow \mu(x,y)$, where $G_{(2)} = \{(x,y) \in G \times G \mid \beta(x) = \alpha(y)\}$ is the set of *composable pairs of* G and an injection $\iota : G \to G, x \to \iota(x)$ called the *inversion map*. We write $x \cdot y$ or xy for $\mu(x,y)$ and x^{-1} for $\iota(x)$.

Definition 1. ([3]) (i) The universal algebra $(G, \alpha, \beta, \mu; G_0)$ is a *semigroupoid*, if μ is *associative*, i.e. $(xy)z = x(yz)$, for all $x, y, z \in G$ such that $(xy)z$ and $x(yz)$ are defined.

(ii) A *monoidoid* is a semigroupoid $(G, \alpha, \beta, \mu; G_0)$ such that the *identities property* holds, i.e. for each $x \in G$ we have $(\alpha(x), x), (x, \beta(x)) \in G_{(2)}$ and $\alpha(x)x = x\beta(x) = x$.

(iii) The 6- tuple $(G, \alpha, \beta, \mu, \iota; G_0)$ is a *groupoid* or a G_0-*groupoid*, if $(G, \alpha, \beta, \mu; G_0)$ is a monoidoid such that the *inverses property* holds, i.e. for each $x \in G$ we have (x^{-1}, x), $(x, x^{-1}) \in G_{(2)}$ and $x^{-1}x = \beta(x)$, $xx^{-1} = \alpha(x)$. □

The definition of the groupoid is equivalent as the one used in [1].

The element $\alpha(x)$ [resp. $\beta(x)$] is the *left unit* [resp. *right unit*] of $x \in G$. The set G_0 is the *unit set* of G and the maps $\alpha, \beta, \mu, \iota$ are the *structure functions* of G.

A G_0- groupoid G such that $|G| = n$ and $|G_0| = m$ is called *finite groupoid of type* $(n; m)$ and it is denoted by $G_{(n;m)}$.

Example 2. (i) *A group G having e as unity, is a $\{e\}$- groupoid with the structure functions given by: $\alpha(x) = \beta(x) = e$, $\iota(x) = x^{-1}$ for all $x \in G$; $\mu(x,y)$ is the product of elements x and y in the group G. Conversely, every groupoid G with one unit is a group.*

(ii) *The pair groupoid $\mathcal{PG}(M)$. The set $\mathcal{PG}(M) = M \times M$ is a $\mathcal{PG}_0(M)$- groupoid with respect to : $\alpha(x,y) = (x,x); \beta(x,y) = (y,y); (x,y)$ and (y',z) are composable iff $y' = y$ and $(x,y) \cdot (y,z) = (x,z)$ and $(x,y)^{-1} = (y,x)$, where $\mathcal{PG}_0(M) = \{(x,x) | x \in M\}$.*

If M is a finite set with $|M| = n$ elements, the groupoid $\mathcal{PG}(M)$ is denoted by $\mathcal{PG}(n)$. □

Definition 3. A *morphism of groupoids* from $(G, \alpha, \beta; G_0)$ into $(G', \alpha', \beta'; G_0')$ is a map $f : G \longrightarrow G'$ such that $f(\mu(x,y)) = \mu'(f(x), f(y))$ for all $(x,y) \in G_{(2)}$. □

Definition 4. Let $(G, \alpha, \beta; G_0)$ be a groupoid. A pair $(H; H_0)$ of nonempty sets such that $H \subseteq G$ and $H_0 \subseteq G_0$ is a *subgroupoid* of G, if $\alpha(H) = \beta(H) = H_0$ and H is closed under multiplication (when it is defined) and inversion.

We say that the subgroupoid $(H; H_0)$ is an *unital subgroupoid* resp.*wide subgroupoid* of G, if $|H_0| = 1$ resp. $H_0 = G_0$. □

Example 5. (i) *Each finite groupoid of type* $(n;1)$ *is a group of order* n. *The groupoid* $\mathcal{PG}(n)$ *is a finite groupoid of type* $(n^2;n)$.

(ii) *The symmetric groupoid* $\mathcal{S}_n$. *A quasipermutation of the set* $M=\{1,2,...,n\}$ *is an injective function* $f_k=\begin{pmatrix} i_1 & i_2 & \dots & i_k \\ f_k(i_1) & f_k(i_2) & \dots & f_k(i_k) \end{pmatrix}$ *where* $1\le k\le n$ *and* $\{i_1,i_2,\dots,i_k\}$ *is an ordered subset of* M. *The set* $\mathcal{S}_n$ *of all quasipermutations has a structure of groupoid of type* $(r;s)$, *where* $r=|\mathcal{S}_n|=\sum\limits_{k=1}^{n}k!(\binom{n}{k})^2$ *and* $s=|\mathcal{S}_{n,0}|=2^n-1$, *see [6].*

$\mathcal{S}_3$ *is a groupoid of type* $(33;7)$. *The subset* $K_{(8;2)}=\{f_j|j=\overline{1,8}\}\subset\mathcal{S}_3$, *where:*

$f_1=\begin{pmatrix}1&2\\1&2\end{pmatrix}, f_2=\begin{pmatrix}1&3\\1&3\end{pmatrix}, f_3=\begin{pmatrix}1&2\\2&1\end{pmatrix}, f_4=\begin{pmatrix}1&2\\1&3\end{pmatrix}, f_5=\begin{pmatrix}1&2\\3&1\end{pmatrix},$

$f_6=\begin{pmatrix}1&3\\1&2\end{pmatrix}, f_7=\begin{pmatrix}1&3\\2&1\end{pmatrix}, f_8=\begin{pmatrix}1&3\\3&1\end{pmatrix}$ *is a subgroupoid of* $\mathcal{S}_3$. □

The *cyclic subgroupoid generated by* a of a groupoid G, denoted by $<a>$, is the intersection of all subgroupoids of G which contain $\{a\}$.

Using the properties of structure functions of a groupoid we have that:

(1.1) $\quad <a>=\{a^n|n\in\mathbf{Z}\}$ if $\alpha(a)=\beta(a)$,

where $a^0=\alpha(a)=\beta(a)$, $a^n=a^{n-1}\cdot a$ for $n\ge 1$ and $a^n=a^{n+1}\cdot a^{-1}$ for $n<0$;

(1.2) $\quad <a>=\{\alpha(a),\beta(a),a,a^{-1}\}$ if $\alpha(a)\neq\beta(a)$.

In the case $\alpha(a)\neq\beta(a)$, the cyclic subgroupoid $<a>$ is a finite groupoid of type $(4;2)$ and it is denoted by $C_{(4;2)}$.

We denote the restrictions of the structure functions α, β, ι and the composition law defined on the groupoid G to $C_{(4;2)}$ by the same symbols. Then the structure functions of $C_{(4;2)}$ are given in the following tables:

x	$\alpha(a)$	$\beta(a)$	a	a^{-1}
$\alpha(x)$	$\alpha(a)$	$\beta(a)$	$\alpha(a)$	$\beta(a)$
$\beta(x)$	$\alpha(a)$	$\beta(a)$	$\beta(a)$	$\alpha(a)$
$\iota(x)$	$\alpha(a)$	$\beta(a)$	a^{-1}	a

μ	$\alpha(a)$	$\beta(a)$	a	a^{-1}
$\alpha(a)$	$\alpha(a)$		a	
$\beta(a)$		$\beta(a)$		a^{-1}
a		a		$\alpha(a)$
a^{-1}	a^{-1}		$\beta(a)$	

A groupoid $(G,\alpha,\beta;G_0)$ is called *cyclic groupoid* if there exists an element $a\in G$ such that $<a>=G$.

Example 6. (i) *The pair groupoid* $\mathcal{PG}(2)=\{p_1=(1,1),p_2=(2,2),p_3=(1,2),p_4=(2,1)\}$ *is a cyclic groupoid isomorphic with* $C_{(4;2)}$.

(ii) *The cyclic groupoid of* $\mathcal{PG}(M)$ *generated by* (a,b) *with* $a,b\in M$ *is* $<(a,b)>=\{(a,a),(b,b),(a,b),(b,a)\}\cong\mathcal{PG}(2)\cong C_{(4;2)}$. *Hence* $\mathcal{PG}(M)$ *is not a cyclic groupoid.* □

It is easy to prove the following theorem.

Theorem 7. *Let* G *be a cyclic* G_0*-groupoid. Then:*

(i) *for* $|G_0|=1\implies G\cong\mathbf{Z}_n$ *when* $|G|=n$ *or* $G\cong\mathbf{Z}$ *when* G *is a infinite;*

(ii) *for* $|G_0|\ge 2\implies G\cong C_{(4;2)}$. □

Lemma 8. Algorithm for determination of cyclic subgroupoids of a groupoid. The program *BGroidAP*6

Let a finite universal algebra $(G,\alpha,\beta,\mu,\iota;G_0)$ such that $|G|=n$ and $|G_0|=m$ with $1\leq m\leq n$, where $G=\{a_1,\cdots,a_m,a_{m+1},\cdots,a_n\}$ and $G_0=\{a_1,\cdots,a_m\}$. We give an algorithm for decide if $(G,\alpha,\beta,\mu,\iota;G_0)$ is a groupoid and for determine the cyclic subgroupoids of G. This algorithm is constituted by the following stages. **Stage I.** *We introduce the initial data*: $n=|G|$, $m=|G_0|$; the functions α,β,ι and μ given by its tables of structure, see [4], [5].

Stage II. *Test for decide if the universal algebra* $(G,\alpha,\beta,\mu,\iota;G_0)$ *considered in the first stage is a groupoid,* see [4], [5]. The element $\mu(a_j,a_k)$ is represented by 0 in the table of input data, if the product $a_j\cdot a_k$ is not defined.

Stage III. *Determine the cyclic subgroupoids of* G. The following steps must be executed:
step 1. Write all nonempty subsets $\{x\}$ of G;
step 2. Determine the cyclic subgroupoid $<x>$ of G generated by $x\in G$;
step 3. Sort by cardinal all cyclic subgroupoids determined in the step 2;
step 4. List the cyclic subgroupoids produced in the above step;
step 5. For each cyclic subgroupoid obtained in the step 4, make its subgroupoid table.

The implementation of the above algorithm on computer is realized in the program *BGroidAP*6, which is composed from the modules *unit*61.*dfm* and *unit*61.*pas*. The module *unit*61.*pas* consists from the principal program followed of procedures and functions.

The principal program of the module *unit*61.*pas* is constituted from the following lignes.

	The module *unit*61.*pas*
01	unit Unit1;
02	interface
03	uses
04	Windows, Messages, SysUtils, Classes,Graphics, Controls, Forms, Dialogs, Grids, DBGrids, ShellAPI, Db, DBTables, StdCtrls, Menus, ExtCtrls, ComCtrls, ToolWin, Spin;
05	const
06	nmax = 200;
07	type
08	TSubSet = Set of Byte;
09	TForm1= class(TForm)
10	MainMenu1: TMainMenu;
11	File1:TMenuItem;
12	OpenFile1: TMenuItem;
13	SaveFile1: TMenuItem;
14	GroupBox1: TGroupBox;
15	StringGrid1: TStringGrid;
16	StringGrid2: TStringGrid;
17	GroupBox2: TGroupBox;
18	StringGrid3: TStringGrid;
19	StringGrid4: TStringGrid;
20	OpenDialog1: TOpenDialog;
21	SaveDialog1: TSaveDialog;
22	Splitter1: TSplitter;
23	Splitter2: TSplitter;
24	ToolBar1: TToolBar;
25	ToolBar2: TToolBar;

	The module *unit*61.*pas*
26	Splitter4: TSplitter;
27	ToolButton1: TToolButton;
28	New1: TMenuItem;
29	ToolBar3: TToolBar;
30	ToolButton4: TToolButton;
31	ToolButton5: TToolButton;
32	Label2: TLabel;
33	SpinEdit1: TSpinEdit;
34	ToolButton6: TToolButton;
35	Label3: TLabel;
36	SpinEdit2: TSpinEdit;
37	ToolButton9: TToolButton;
38	Savesubgroupoid1: TMenuItem;
39	StatusBar1: TStatusBar;
40	StatusBar2: TStatusBar;
41	ToolButton3: TToolButton;
42	ToolButton11: TToolButton;
43⋆	procedure FormShow(
	Sender: TObject);
44⋆	procedure Button2Click(
	Sender:TObject);
45⋆	procedure StringGrid1SetEditText(
	Sender:TObject; ACol,ARow:
	Integer; const Value: String);
46⋆	procedure StringGrid2SetEditText(
	Sender: TObject; ACol, ARow:
	Integer; const Value: String);
47⋆	procedure OpenFile1Click(
	Sender: TObject);
48⋆	procedure SaveFile1Click(
	Sender: TObject);
49⋆	procedure StringGrid3SelectCell(
	Sender:TObject; ACol, ARow:
	Integer; var CanSelect: Boolean);
50⋆	procedure New1Click(
	Sender: TObject);
51⋆	procedure ToolButton4Click(
	Sender: TObject);
52	procedure ToolButton9Click(
	Sender: TObject);
53⋆	procedure Savesubgroupoid1Click(
	Sender:TObject);
54⋆	procedure ToolButton3Click(
	Sender: TObject);

	The module *unit*61.*pas*
55	private
56	ForcedStop :Boolean;
57	err_message : String;
58	subgr : array[1..10000] of TSubSet;
59	units, SelectedSub : TSubSet;
60	m, n, nsub : Integer;
61	h :array[0..nmax, 0..nmax] of Byte;
62	u_left,u_right, inv : array[0..nmax]
	of Integer;
63⋆	procedure WMDropFiles(var Msg:
	TWMDropFiles);
	message WM_DROPFILES;
64⋆	procedure PerformFileOpen(
	const FileName1 :string);
65⋆	procedure PerformFileSave(
	const FileName1 : string);
66⋆	procedure PerformSaveSubgroupoid(
	t : TSubSet; const FileName1 : string);
67⋆	procedure MakeUnitsTable;
68⋆	procedure MakeGroupoidTable;
69⋆	procedure MakeSubgroupoidTable(
	t : TSubSet);
70⋆	function ToStr(x : Integer) : String;
71⋆	function SubsetToString(t
	: TSubSet) : String;
72⋆	function Cardinal(t : TSubSet) : Byte;
73⋆	procedure Cover(var t : TSubSet);
74⋆	function AlreadyFound(t
	: TSubSet) : Boolean;
75⋆	procedure AddSubgroupoid(t : TSubSet);
76	procedure GenerateCyclics;
77⋆	procedure SortByCardinal;
78⋆	procedure ListSubgroupoids;
79⋆	function IsStructure : Boolean;
80⋆	function IsSemigroupoid : Boolean;
81⋆	function IsMonoidoid : Boolean;
82⋆	function IsGroupoid : Boolean;
83	public
84	end;
85	var
86	Form1: TForm1;
87	implementation
88	{$R *.DFM}
89	end.

The procedures and functions marked by the symbol "$\star$" can be find in the program *BGroidAP*2 , see [5]. The *procedure TForm1.GenerateCyclics;*, denoted by procedure 1, is formed by the lignes $(1)01-(1)11$ and the *procedure TForm1.ToolButton9Click(Sender: TObject);*, denoted by procedure 2, is formed by the lignes $(2)01-(2)08$. These lignes are presented in the following table:

	procedure 1 and procedure 2
(1)01	var
(1)02	i : Byte;
(1)03	t : TSubSet;
(1)04	begin
(1)05	for i := 1 to n do begin
(1)06	t := [i];
(1)07	Cover(t);
(1)08	if not AlreadyFound(t) then
(1)09	AddSubgroupoid(t);
(1)10	end;

	procedure 1 and procedure 2
(1)11	end;
(2)01	begin
(2)02	nsub := 0;
(2)03	ForcedStop := false;
(2)04	GenerateCyclics;
(2)05	SortByCardinal;
(2)06	ListSubgroupoids;
(2)07	StatusBar2.SimpleText := tostr(nsub) + ' subgroupoid(s) found.'
(2)08	end;

Example 9. *Determination of cyclic subgroupoids of the groupoid $K_{(8;2)}$(see Example 1.2(ii)). Using the correspondence $K_{(8;2)} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\} \longleftrightarrow \{1,2,3,4,5,6,7,8\}$ the input data are the following:*

$$8$$
$$2$$
$$\begin{array}{cccccccc} 1 & 2 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 2 & 1 & 1 & 2 \\ 1 & 2 & 3 & 6 & 7 & 4 & 5 & 8 \end{array}$$

$$\begin{array}{cccccccc} 1 & 0 & 3 & 4 & 5 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 6 & 7 & 8 \\ 3 & 0 & 1 & 5 & 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 1 & 3 & 5 \\ 0 & 5 & 0 & 0 & 0 & 3 & 1 & 4 \\ 6 & 0 & 7 & 2 & 8 & 0 & 0 & 0 \\ 7 & 0 & 6 & 8 & 2 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 7 & 6 & 2 \end{array}$$

Execute the program BGroidAP6 for $K_{(8;2)}$ and the window program of obtained results is presented in the Figure 1.

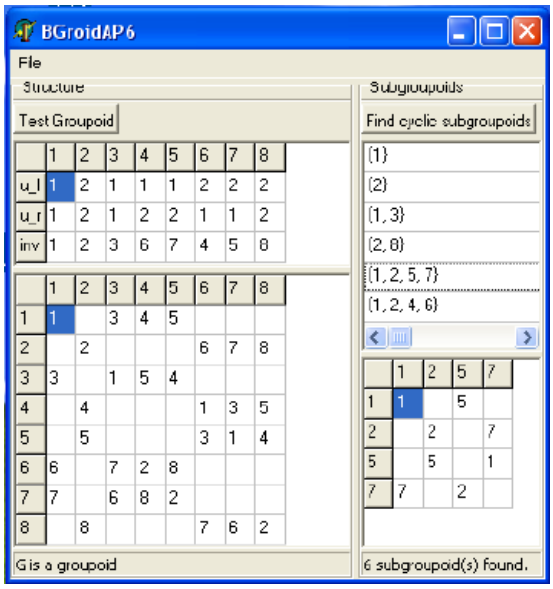


Figure 1: The cyclic subgroupoids of a $(8;2)$-groupoid

Therefore, $K_{(8;2)}$ is a groupoid and it has the following 6 cyclic subgroupoids (see, Figure 1): $C^1_{(1;1)} = \{f_1\}, C^2_{(1;1)} = \{f_2\}, C^3_{(2;1)} = \{f_1, f_3\}, C^4_{(2;1)} = \{f_2, f_8\}, C^5_{(4;2)} = \{f_1, f_2, f_4, f_6\}, C^6_{(4;2)} = \{f_1, f_2, f_5, f_7\}$.

We have that the cyclic subgroupoids $C^3_{(2;1)}, C^4_{(2;1)}$ are isomorphic with $\mathbf{Z}_2$ and the cyclic subgroupoids $C^5_{(4;2)}, C^6_{(4;2)}$ are isomorphic with $C_{(4;2)}$. □

Example 10. *Applying the program BGroidAP6 we have that the dihedral group $D_6 = \{e, a, a^2, a^3, a^4, a^5, b, ab, a^2b, a^3b, a^4b, a^5b | a^6 = e, b^2 = e, ba = a^5b\}$ has 10 cyclic subgroups: $C_1 = \{e\}, C_2 = \{e, a^3\}, C_3 = \{e, b\}, C_4 = \{e, ab\}, C_5 = \{e, a^2b\}, C_6 = \{e, a^3b\}, C_7 = \{e, a^4b\}, C_8 = \{e, a^5b\}, C_9 = \{e, a, a^2, a^4\}, C_{10} = \{e, a, a^2, a^3, a^4, a^5\}$.*

We have that $C_i \cong \mathbf{Z}_2$ for $i = \overline{2,8}, C_9 \cong \mathbf{Z}_3$ and $C_{10} \cong \mathbf{Z}_6$. □

Remark 11. In the paper [4] is given the program *BGroidAP*3 for determination of the wide subgroupoids of a groupoid. Also, in the paper [2] is presented the program *BGroidAP*8 for determination of the unital subgroupoids of a groupoid. □

References

[1] A. Coste, P. Dazord and A. Weinstein, *Groupoides symplectiques,* Publ. Dept. Math. Lyon, 2/A (1987),1-62.

[2] M. Degeratu, *The program BGroidAP*8 *for computing of unital subgroupoids of groupoid,* An. Univ. de Vest din Timişoara, 2005, to appear.

[3] Gh. Ivan, *Algebraic constructions of Brandt groupoids,* Proceedings of the Algebra Symposium, " Babeş- Bolyai" University, Cluj, (2002), 69-90.

[4] Gh. Ivan, M. Ivan,*The program BGroidAP*3 *for determination of wide subgroupoids of a Brandt groupoid,* Proceededings of ICCC 2004, Băile Felix Spa-Oradea, Romania, 2004, 209-213.

[5] Gh. Ivan, G. Stoianov, *The program BGroidAP*2 *of determination of the subgroupoids of a groupoid,* An. Univ. de Vest din Timişoara, Seria Mat.- Inform., vol **42**, fasc. 1, 2004, 93-117.

[6] M. Ivan, *General properties of the symmetric groupoid of a finite set,* An. Univ. de Vest din Timişoara, Seria Mat.- Inform., vol **42**, fasc. 1, 2004, 93-117.

[7] A. Weinstein, *Groupoids: Unifying Internal and External Symmetries,* Notices Amer. Math. Soc., 43 (1996), 744 - 752.

Marian Degeratu
University of Oradea
Department of Mathematics and Computer Science
Address: 1, Universităţii St.,410087, Oradea, Romania
E-mail: mariand@uoradea.ro

Gheorghe Ivan
West University of Timişoara
Department of Mathematics
Address: 4, Bd. V. Pârvan, 300223, Timişoara, Romania
E-mail: ivan@math.uvt.ro

Mihai Ivan
West University of Timişoara,
Department of University Colleges
Address: 4, Bd. V. Pârvan, 300223, Timişoara, Romania
E-mail: ivan@math.uvt.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 187-192

Convex cost flow. Adaptation of network simplex algorithm

Cristian Dobre

Abstract: Starting from the optimality test in Network Simplex Algorithm with linear objective function and the way of keeping a convex function defined on integer numbers, is proposed an adaptation of the algorithm mentioned above. The main topics in this article are related to the way of finding optimality conditions in the convex model, together with a description of the adapted algorithm.

Keywords: convex programming, combinatorial optimization, optimality conditions, network simplex.

1 Introduction

Let G=(N,A,c,u) be a network, where N denotes the set of nodes, A represents the set of arcs, c is the cost function and u is the capacity function. If we denote by x the coresponding flow function, the liniar model of the minimum cost flow problem is the following:

$$\begin{aligned} &\min\left(z=\sum_{(i,j)\in A} c_{ij}x_{ij}\right)\\ &\sum_{j/(i,j)\in A} x_{ij}-\sum_{j/(j,i)\in A} x_{ji}=b(i) \qquad \forall i\in N\\ &0\le x_{ij}\le u_{ij} \qquad \forall (i,j)\in A \end{aligned} \tag{1}$$

Without loss of generality, we have considered zero lower bounds for the flow x and also the model has only one source node **s** and one sink node **t**, therefore the node values are b(i)=0, $i \in N$ except for s and t.

In this model, if we denote $g_{ij}(x_{ij}) = c_{ij}x_{ij}, \forall (i,j)\in A$ where $g_{ij} : D_{ij} \longrightarrow \mathtt{N}$ and $D_{ij} = [0,u_{ij}] \cap \mathtt{N}$ then we can rewrite the objective function as $z = \sum_{(i,j)\in A} g_{ij}(x_{ij})$

The purpose of this article is to extend the linear form of the functions g_{ij} and consider them as convex functions of variable x_{ij}.

The input data of the linear model requires 2m values,m for arc costs and m for arc capacities, where m is the number of arcs in set A. Unlike the linear one, the convex model requires O(mu) values for the input data,where u = $max_{(i,j)\in A} u_{ij}$, because we need to introduce u_{ij} costs for every arc (i,j). Even if we consider the concise model of the function we must first compute the costs mentioned above in O(mu) time. This situation occurs because we have a different cost for each unit of flow sent along an arc. (See example 1).

If we have the concise function model, we compute the cost of sending the k-th unit of flow on arc (i,j) with the formula:

$$c_{ij}^k = \frac{g_{ij}(k)-g_{ij}(k-1)}{k-(k-1)} = g_{ij}(k)-g_{ij}(k-1) \tag{2}$$

We solve the problem assuming that the solution(vector x) is contiguous [3], that is, if for the k-th unit of flow we have the cost c_{ij}^k then for the next one we have the cost c_{ij}^{k+1}.

Example 1. *Let (i,j) be an arc form the set A, with the capacity u_{ij}=3. Notice the difference between the linear function $g_{ij}(x_{ij}) = 2x_{ij}$ and the nonlinear one $g_{ij}(x_{ij}) = 2x_{ij}^2$ on the same $D_{ij} = [0,3] \cap N$ (Figure 1).*

In the graphics, the slope of each segment represents the cost of sending one more unit of flow on arc (i,j).

If the linear model makes use of the constant costs on each segment(the same slope) in order to establish the optimality conditions, the convex model(proposed in my article) will develop a similar theory, and this is the main target of the paper. To sum up, the convex model for which is proposed an adaptation of Network Simplex Algorithm is:

$$\begin{aligned} &\min\left(z=\sum_{(i,j)\in A} g_{ij}(x_{ij})\right)\\ &\sum_{j/(i,j)\in A} x_{ij}-\sum_{j/(j,i)\in A} x_{ji}=b(i) \qquad \forall i\in N\\ &0\le x_{ij}\le u_{ij} \qquad \forall (i,j)\in A \end{aligned} \tag{3}$$

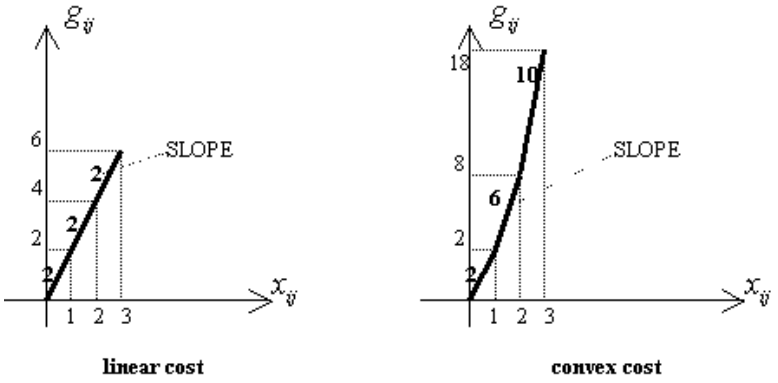


Figure 1: Flow cost comparison

where $g_{ij}(x_{ij}) : [0,u_{ij}] \cap N \longrightarrow Z$ are convex functions.

2 Optimality Conditions

I present here a brief description of the network simplex algorithm for the problem (1).A detalied description of the algorithm can be found in [1]. The primal network simplex algorithm maintains a basis structure (T,L,U) which is primal feasible (all constraints are satisfied), but dual infeasible. A dual solution of the minimum cost flow problem is a vector π of node potentials. For a given dual solution π, we define the reduced cost of an arc (i,j) as $c^{\pi}_{ij} = c_{ij} - \pi(i) + \pi(j)$. The basis structure (T,L,U)is called dual feasible if there exists a set of node potentials π satisfying the following optimality conditions $c^{\pi}_{ij} = 0$ for all $(i,j) \in T, c^{\pi}_{ij} \geq 0$ for all $(i,j) \in L$ and $c^{\pi}_{ij} \leq 0$ for all $(i,j) \in U$. The basis is said to be optimal if and only if it is both feasible and dual feasible, therefore dual feasibility is equivalent with optimality conditions. T,L and U are all subsets of A. T is a particular one consisting in exact n-1 arcs (where n is the number of nodes in N) that form a spanning tree for the given network. We compute the vector π by solving the system $c^{\pi}_{ij} = 0$ for all $(i,j) \in T$.I intend to give another interpretation to the values of π, interpretation that will help to elaborate optimality conditions for the convex model.

Remark 2. In the present and following sections I will refer to the subset $T \subset A$ as (optimal) spanning tree solution.

The potential vector π was introduced in order to characterize the cost of transporting **one** unit of flow from node s to each node in N using arcs in T. The cost of transporting one unit of flow from node s to node k is obtained over undirected paths starting with $\pi(s) = 0$ and then substracting the cost c_{ij} if the arc is a forward arc in the path, or adding the cost c_{ij} if the arc is a backward one. When we reach node k the value obtaind by the algorithm mentioned above is exactly $\pi(k)$. The same values we obtain by solving the system $c^{\pi}_{ij} = 0$ for all $(i,j) \in T$. Optimality can be interpreted in the following fashion: suppose we select an arc $(p,q) \in L$, we compare if the cost of sending one unit of flow from p to q using undirected paths in T, which is **$(\pi(p) - \pi(q))$**is better than the cost we have on arc (p,q) that is **(c_{pq})**. The same interpretation can be obtained for the arcs in U. Here is a numerical example for the statement above:

Example 3. *In figure **2A**, the value on each arc represents the cost of transporting one unit of flow and in figure **2B** we have the coresponding spanning tree solution,T(for details obtaining this spanning tree see [1]). Using costs on arcs in T and the rule mentioned in the previous paragraph we obtain $\pi = (0,-1,6,5,-3,2)$. The same values we can compute from $c^{\pi}_{ij} = 0 for all (i,j) \in T$. More, if arc (3,5) would have $x_{35} = 0$ then we can compare the cost of transporting one unit of flow on arc (3,5) which is $c_{35} = 6$ with the cost of transporting the same unit using the arcs in T, which is $\pi(3) - \pi(5) = 9$. Obviously is more efficient to use arc (3,5) therefore optimality condition is not satisfied.*

The most important remark is that a similar reasoning it is not posible for the convex model because the cost changes at every unit of flow that we send forth or back on every arc in the network(if the cost function is not linear). The result is the following: if the arc (p,q) has the capacity u_{pq} and the flow $x_{pq} = k$ we should keep two cost values, namely the cost of transporting the $(k+1)^{th}$ unit, (c^{k+1}_{pq}) and the cost of sending back the k^{th} unit,(c^{k}_{pq}). If $x_{pq} = 0$ we set $c^{0}_{pq} = -\infty$ and if $x_{pq} = u_{pq}$ we set $c^{u_{pq}+1}_{pq} = \infty$.

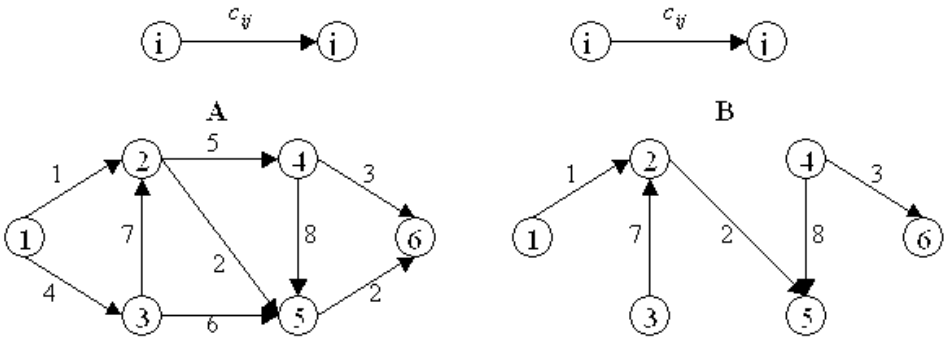


Figure 2: Flow cost comparison

Definition 4. Let c^d_{pq} be ***the forward reduced cost*** of an arc (p,q), that is: the cost of transporting one unit of flow in the cycle formed with the arcs in T and arc (p,q) when the orientation of the cycle is the same as the orientation of arc (p,q).

Let c^i_{pq} be ***the backward reduced cost*** of an arc (p,q), that is: the cost of transporting one unit of flow in the cycle formed with the arcs in T and arc (p,q) when the orientation of the cycle is oposite to the orientation of arc (p,q).

Theorem 5. *A spanning tree solution is optimal if and only if* $\forall(i,j) \in A-T$ *we have* $c^d_{ij} \geq 0$ *and* $c^i_{ij} \geq 0$. *The flow on all arcs in A, coresponding to this tree is the optimal solution for the minimum convex cost flow problem (3).*

Proof. Let $x^* = (x^*_{ij})_{(i,j)\in A}$ be the optimal solution. Assume there is $(k,l) \in A$ with $c^d_{kl} < 0$ or $c^i_{kl} < 0$. Then by sending one unit of flow in the cycle formed with arc (k,l) and arcs in T the total cost of transport decreases by $|c^{d/i}_{kl}|$ units, therefore x^* can not be optimal.

Reverse, if $c^d_{ij} \geq 0$ and $c^i_{ij} \geq 0\ \forall(i,j) \in A-T$ and assume that the curent solution, x^* is not optimal, then $\exists x^0$ optimal flow obtained by modifying $\vec{x^*}$ with at least one unit. Let W be the cycle where the flow changing occurs. Since W is formed with arcs from $T \cup (i,j)$ and $c^d_{ij} \geq 0$ and $c^i_{ij} \geq 0$ the total cost of transportation will increase(or remain the same) no matter what is the orientation of W. So either coresponding value of the objective function z for x^0 is the same or x^0 is not optimal.

3 Adaptation of the algorithm

We saw in the previous section, which are the optimality conditions. For this algorithm we only use the set T representing arcs in A that form a spanning tree for the network. If at a certain moment the optimality conditions are satisfied then the algorithm terminates, and we call T optimal spanning tree solution(see remark 2). On the other hand, if T is not optimal we select the arc (p,q) having one of the reduced costs(forward or backward) given by the formula:

$$c^{d/i}_{pq} = \max_{(i,j)\in A-T} \{ \max_{c^d_{ij}<0, c^i_{ij}<0} |c^d_{ij}|, |c^i_{ij}| \} \tag{4}$$

Arc (p,q) is called **entering arc** and will form a cycle(denoted by W) togheter with arcs in T. There are two situations: entering arc was chosen because of its forward reduced cost or because of its backward reduced cost. In the first case we define the orientation of W the same as the orientation of (p,q), and in the second one, oposite to the orientation of (p,q). Algorithm sends **one unit** of flow along the orientation of W. After each unit of flow modified in the network we have different costs on arcs(see example 1).

In order to avoid testing optimality at each unit flow changing we could use binary search, in the way it is presented in [1] for the Cycle canceling algorithm.

After we have modified the flow, in case one of the arcs in W reaches its inferior or superior flow limit then he will be **the leaving arc**. If any of the arcs satisfy the condition above we select as leaving arc, the arc (r,s)for which $c^k_{rs} + c^{k+1}_{rs}$ is maximum, where k is the amount of folw sent on arc (r,s). This rule assure that, if necessary the same arc will enter the structure T and algorithm will continue sending flow on the same cycle.(Notice that costs change every time flow changes so we are not positive that sending all posible flow on arc is an optimal decision).

NETWORK SIMPLEX CONVEX COST ALGORITHM
BEGIN

- find a maximum flow in network G
- find arcs in set T
- **WHILE** $\exists \quad c_{ij}^d < 0$ or $c_{ij}^i < 0$ for $(i,j) \in A - T$
  - select entering arc (p,q)
  - push one unit of flow along the orientation of cycle W
  - update arcs flow and costs
  - identify the leaving arc (r,s)
- **END**

END.

Theorem 6. *Algorithm* ***Network Simplex Convex Cost*** *determines the minimum cost flow for the network (N,A,c,u).*

Proof. The algorithm starts with a maximum flow in network G. The amount of flow sent from source node s to sink node t is constant. Algorithm modifies flow along cycles. The minimun cost of the objective function is obviously inferior limited. Starting from a certain value of function z, given by the maximum flow determined in line 1 algorithm decreases the value of z by modifying flow along cycles with negative reduced costs.When no such cycles can be determined for arcs in A-T then algorithm terminates, and the value of z can not decrease anymore.

Remark 7. Regarding algorithm's complexity, this depends on the way of solving maximum flow procedure; and also the bottleneck operation is the number of the execution for cycle WHILE, which is obviously pseudopolynomial (depends on arc capacities). Also the input data is pseudopolinomyal(for every arc we must determine each cost of sending one more unit of flow) as I mentioned in example 1. Parallel solutions can be found in order to determine leaving arc rule [4], or even for the input cost data, but this could be a topic for my next papers.

4 Numerical example

Let us consider the numerical example from figure 3, where $g_{ij}(x_{ij}) = c_{ij}x_{ij} + x_{ij}^2$. We have there the values for the coeficients of the cost functions c_{ij} and arcs capacities u_{ij} (A), a maximum flow distribution (determined with any flow algorithm) x_{ij} (B) and a spanning tree (C). Arcs in T, representing spanning tree solution, were chosen from the arcs with $0 < x_{ij} < u_{ij}$ and if there are not n-1 arcs satisfying this condition, the rest were chosen at random (under restriction to form a spanning tree). Therefore we have T={(1,2),(2,4),(3,4)}.

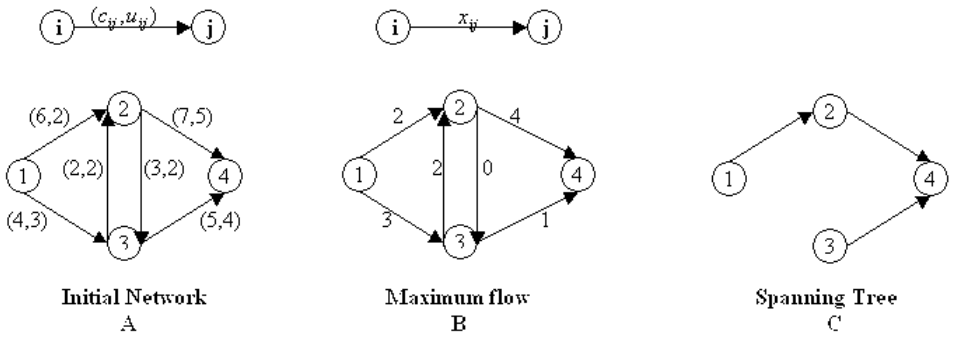


Figure 3: Initial values

We compute the costs of transporting each unit of flow on every arc in netwwork using formula 2, and present them in figure 4.

x_{12}	1	2
$g_{12}(x_{12})$	7	16
slope	7	11

x_{23}	1	2
$g_{23}(x_{23})$	4	10
slope	4	6

x_{24}	1	2	3	4	5
$g_{24}(x_{24})$	8	18	30	44	60
slope	8	10	12	14	16

x_{32}	1	2
$g_{32}(x_{32})$	3	8
slope	3	5

x_{13}	1	2	3
$g_{13}(x_{13})$	5	12	21
slope	5	7	9

x_{34}	1	2	3	4
$g_{34}(x_{34})$	6	14	24	36
slope	6	8	10	12

Figure 4: Cost values / arc

Figure 5 A shows the costs c_{ij}^{k+1} and c_{ij}^{k} for each arc (i,j) in the network, where k is the amount of flow sent through arc (i,j).These costs follow the definition in section 2. Next we need to compute the reduced costs for arcs in A-T using definition 4. Notice that for arcs with $x_{ij} = 0$ we don't have to compute c_{ij}^{i} and for arcs with $x_{ij} = u_{ij}$ we don't have to compute c_{ij}^{d} because both are ∞.

A-T={ (1,3),(3,2),(2,3)} and $c_{13}^{i} = -c_{13}^{3} + c_{12}^{3} + c_{24}^{5} - c_{34}^{1} = -9 + \infty + 16 - 6 > 0$. We notice that for the backward arcs in W=(1,2,4,3,1)we substract c_{ij}^{k} and for the forward arcs we add c_{ij}^{k+1}, where k, as I said before, is the curent amount of flow sent through arc (i,j). In fact c_{13}^{i} is a comparison between the cost directly on arc (1,3) and cost along existing undirected path (1,2,4,3). This is **the same** ideea as in the linear model, that led me to this optimality test. In the same way we compute: $c_{32}^{i} = -5 + 8 - 14 = -11 < 0$ and $c_{23}^{d} = 4 + 8 - 14 = -2 < 0$. The entering arc is (3,2), W=(3,4,2,3)in oposite orientation of arc (3,2).Flow and costs after sending one unit of flow along W are presented in figure 5 B and 5 C respectivly. Bold dashed arcs form the new spanning tree solution, after determining the leaving arc (2,4) as a consequence of $\max\{14+12, 5+3, 10+8\}$

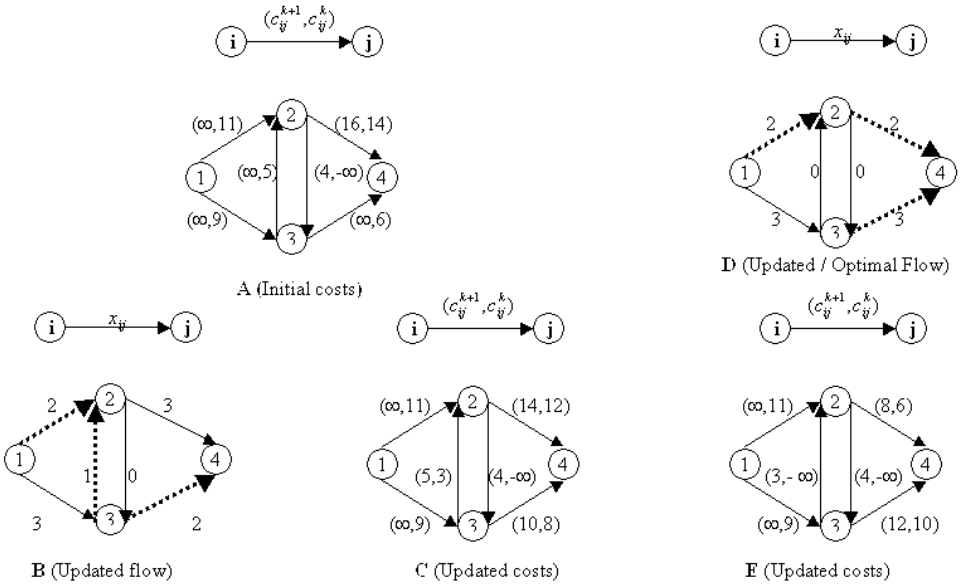


Figure 5: Numerical example

Next iteration we have: T={ (1,2),(3,2),(3,4) }. A-T={ (1,3),(2,4),(2,3) }. $c_{13}^{i} = -9 + \infty - 3 > 0$, $c_{24}^{i} = -12 - 3 + 10 < 0$, $c_{24}^{d} = 14 - 8 + 5 > 0$ and $c_{23}^{d} = 4 + 5 > 0$. Entering arc is (2,4) and W=(4,2,3,4) in oposite orientation of arc (2,4).Flow and costs after sending one unit of flow along W are presented in figure 5D and 5 E respectivly. Bold dashed arcs form the new spanning tree solution, after determining the leaving arc (3,2) as a consequence of $x_{32} = 0$.

Next iteration we have: T={ (1,2),(2,4),(3,4) }. A-T={ (1,3),(3,2),(2,3) }. $c_{13}^{i} = -9 + \infty + 8 - 10 > 0$, $c_{23}^{d} = 4 + 12 - 6 > 0$ and $c_{32}^{d} = 3 + 8 - 10 > 0$. Optimality conditions are satisfied, so the algorithm ends with optimal

flow in figure 5 D and optimal value of z=16+21+18+24=79.

References

[1] R.K. Ahuja,T.L.Magananti & J.B.Orlin, “Network Flows: Theory, Algorithms and Aplications”, *Prentice Hall, Englewood,Cliffs, New Jersey*,1993.

[2] D.P.Bertsekas, “Linear Network Optimization”, *The MIT Press, London, England*, 1991.

[3] R.K. Ahuja,Dorit S. Hochbaum & J.B.Orlin, “A Cut-Based Algorithm for the Nonlinear Dual of the Minimum Cost Network Flow Problem ”, *Submitted for publication to Mathematical Programming*,2003.

[4] C.Dobre, “Network Simplex Algorithm.Parallel solution for finding the leaving arc”, *Bulletin of Transilvania University of Brasov* , 2005.

Cristian Dobre
Transilvania University of Brasov
Department of Computer Science
Address: 50, Iuliu Maniu St., Brasov, Romania
E-mail: cr.dobre@unitbv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 193-198

WinNet - a network tool

Sanda Dragos, Radu Dragos

Abstract: Building Internet-like network topologies and performing simulations and/or emulations on them is a task that all network-related topic researchers came across during their study. We propose here a new tool which incorporates network topology generation, visualization, simulation and an analyzing mobile agent based tool. This application can be used for research purposes as well as an educational instrument. Even if it is mainly aimed for computer networks, this application can be also used to solve non-computer network related problems such as logical tasks or database searches.

Keywords: Network Simulator, topology generator, mobile agent, Wave

1 Introduction

The explosive growth of the Internet has been accompanied by a wide range of problems ranging from routing to resource reservation or administration. New algorithms and policies that try to solve such problems need to be tested on an abstraction network or on an actual network. Researchers rarely use real networks as their testbeds because networks large enough to be representative are also very expensive and difficult to control. A more efficient way to perform the tests is on emulated networks.

At first, researchers used very simple topologies (e.g. mesh, star, tree, ring, lattice, grid, cube) for their simulations. However, they do not reflect any real network, and are generally used now only to simulate specific scenarios such as LANs or other shared communication media. Simulating Internet-like network topologies is a very difficult task as the Internet does not fit into some specific formats, and it is constantly changing. Thus, topology generators were developed. There are a wide variety of network topology generators available. Waxman [1] developed one of the first topology generators, which is concerned with general random networks. One of the most popular topology generators available is GT-ITM [2]. It focuses on reproducing the hierarchical structure of the topology of the Internet, but also includes about five types of flat random graphs. Tiers [3] is based on a three-level hierarchy aimed at reproducing the differentiation between Wide-Area, Metropolitan-Area and Local-Area networks comprising the Internet. Inet [4] and PLRG [5] are two topology generators concerned with emulating the connectivity properties of Internet topologies as reported in [6].

Network simulators use topology generators like those mentioned above. For instance, the Network Simulator (NS) [7] uses Inet, GT-ITM and Tiers topology generators, along with topologies specified by hand, while OMNeT ++ [8] uses Inet. Most network simulators have already implemented existing data-link, network and transport protocols (e.g. IP, TCP, UDP, PPP, Ethernet, MPLS with LDP and RSVP-TE).

In this context we propose a new simulation tool, called WinNet, that can be also used as a planning instrument and topology analyzer. It uses GT-ITM as a network topology generator and Network Simulator (NS) as a simulating instrument. It also creates, by using a mobile agent system called WAVE, “virtual” networks on which to perform numerous analyzing tasks.

The Wave technology [9, 10] is based on parallel spreading of recursive program code (or *waves*) in *open systems*[1], accompanied by dynamic creation of virtual Knowledge Networks (KNs). Such networks can persist and reflect any declarative or procedural information. Moreover, they may become active and capable of self-evolution, self-organization and self-recovery. Other *waves* can navigate, control and modify KNs. All these actions are performed without a central memory or a centralized control. Another important Wave feature is that routines such as synchronization, message passing and garbage collection are implemented within the Wave Interpreter which resides on physical nodes rather than being implemented within mobile agents as with other mobile agent technologies. This and its syntax make Wave code very compact, perhaps 20 to 50 times shorter that equivalent programs written in CC++ or Java [11].

[1] Constantly evolving and changing in time and space, while intensively exchanging information with other systems and with the environment [9]. Examples of open system are computer networks.

2 The WinNet tool

WinNet is a Linux-based application that creates random network topologies by using GT-ITM on which it may perform network simulations using NS by automatically building a network simulating application written in Tk/Tcl. It extends GT-ITM by associating different (single or multiple) costs to each link. Such costs are randomly chosen from given intervals. The most important WinNet feature, however, is that, by using a mobile agent system called WAVE, it is able to create and modify virtual networks having specified topologies and to perform analyzing tasks (such as finding all articulation points, determining the diameter, finding all and the maximum cliques) on these topologies.

The WinNet name is formed of two particles starting with capitals: **Win** and **Net**. The **Win** particle contains the initials of the three instruments used: **W**AVE, GT-**I**TM and **N**S, while **Net** indicates that it is an networking tool.

The application starts with a selection window depicted in Fig. 1 which allows the selection of any starting stage. That is because files created with this application during any previous utilization can be also used.

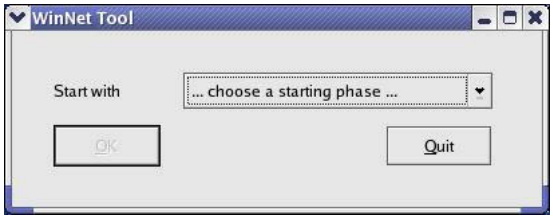
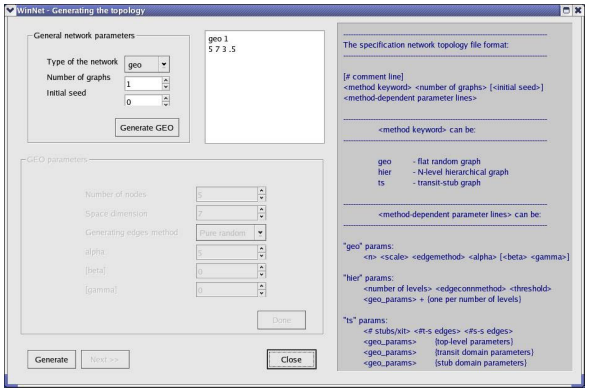
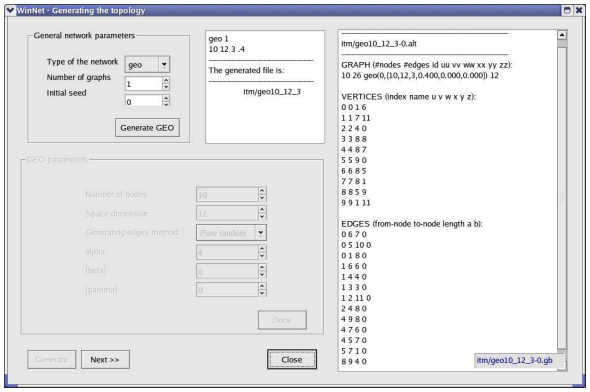


Figure 1: The main application window

The first application stage consists in building the random topology using GT-ITM. A very short help page is also provided on the right hand side of the window (see Fig. 2(a)). Selecting values for variables that may exist only within specific intervals are implemented by using combo-boxes. They also facilitate the creation of the topology description file, which can be seen and modified if needed in a textbox environment.



(a) initial phase



(b) final phase

Figure 2: Generating a random topology

After specifying the topology description file, the topology can be created by pressing the “Generate” button. A configuration file specifies the number (one or more), the type (additive, multiplicative, concave) and the order (increasing, decreasing) of metrics to be associate with each link. Their accepted value intervals can also be specified in the configuration file. The newly created file specifying the generated topology can be viewed on the right hand side of the same window (see Fig. 2(b)) in the exact place previously occupied by the help page. The name of the topology file is generated using information from the topology specification file (i.e. *<type of the network><number of nodes>_<space dimension>_<method used for generating edges>*) and it is listed on the right-down corner of the textbox containing the content of the same file. If more files are generated, they all will be listed there.

The next application stage is converting the random topologies created in the previous stage into WAVE and NS programs as depicted in Fig. 3. Random topology files generated using any previous utilization of the WinNet application can also be used and are accessible for selection. The application allows the user to select either the Wave or the NS conversion or both of them. Such conversion will apply to all random topology files existent in the *selected files* environment. The code generated for WAVE and/on NS programs is listed in the right hand side of this interface window.

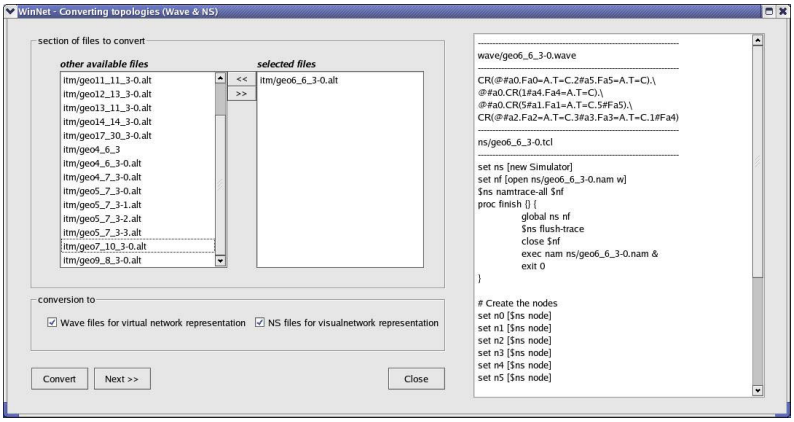
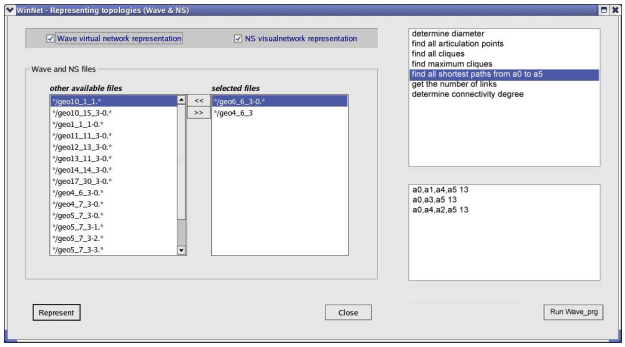
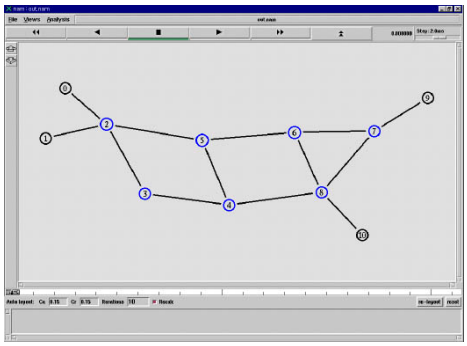


Figure 3: Converting random topologies into Wave and/or NS programs

The third application stage is the Wave and/or NS representations. An interface window (see Fig. 4(a)) allows choosing a specific topology to be represented either visually by NS, or virtually by WAVE. By using NS the user would be able to see the network topology as presented in Fig. 4(b) and to watch the packet flows over different scenarios that may be specified in the configuration file. Such scenarios may include setting an TCP or UDP connection, setting LSP's[2] over different paths, watch the packet queues and observe if packets are dropped or not, and so on. This scenarios can also be correlated with the findings offered by the Wave programs. One such example is to set up LSP's over shortest paths.



(a) Representation interface and Wave programs selection



(b) Network visualization using NS

Figure 4: Representation of random topologies

Selecting the WAVE representation results in creating a virtual network on one or more physical network nodes. Such network contains specified information associated with each link (as presented previously on the first application stage). The virtual network will reside on the nodes where it was created and it can be navigated by small mobile agents, called *waves*, programmed to perform a very large range of operations. Examples of such operations that can be performed by *waves* implemented by WinNet are described below and can be viewed in Fig. 4(a).

determining the diameter - The diameter is the maximum distance[3] between any two network nodes. Such

[2] *Label Switched Path* used by the Multi-Protocol Label Switching (MPLS).

[3] The *distance* between two nodes is the number of nodes to be traversed in order to reach from one node to the other.

metric might be used in estimating Time-to-Live (TTL)-like variables used by different protocols.

finding all articulation points - An articulation point is a node which if removed divides the initial graph. The number and the exact location of articulation points can help in avoiding congestion points. Moreover, if more adjacent nodes are articulation points, their common link may become over-utilized.

finding all cliques - A clique is any complete (full-mesh) subgraph/topology within the initial topology. Knowing this information is useful for determining and avoiding possible cycles.

all shortest paths between two nodes - *Shortest*[4] paths can be determined based on the metrics associated with links. The path computation is performed in a parallel and distributed manner using *waves*. This allows determining "shortest" paths based on multiple metrics. Shortest path trees starting from one node can also be easily determined. They are mainly used in multicast routing.

get the number of the links - This feature can be used as feedback in order to determine how different GT-ITM configuration files can affect the generated network topology in terms of number of links.

determine connectivity degree - The connectivity degree is twice the number of links over the number of nodes and represents an average of the node degree, meaning the average number of incident links to a node. This feature is important, for instance, in mobile agent programming because every mobile agent may multiply within every node with the number of incident links, and thus in highly connected networks they might generate too much traffic.

Other non-computer networks related problems, but which can be represented in a network-like model, can also be solved using this application. The configuration file can specify the assignment of non-numerical values to each link and/or node. Example of such applications are logic tasks and parallel database searches.

The WinNet flowchart is depicted in Fig. 5. It presents the main stages of our application, the files used, the places where the configuration file is consulted and the dual finality of this application.

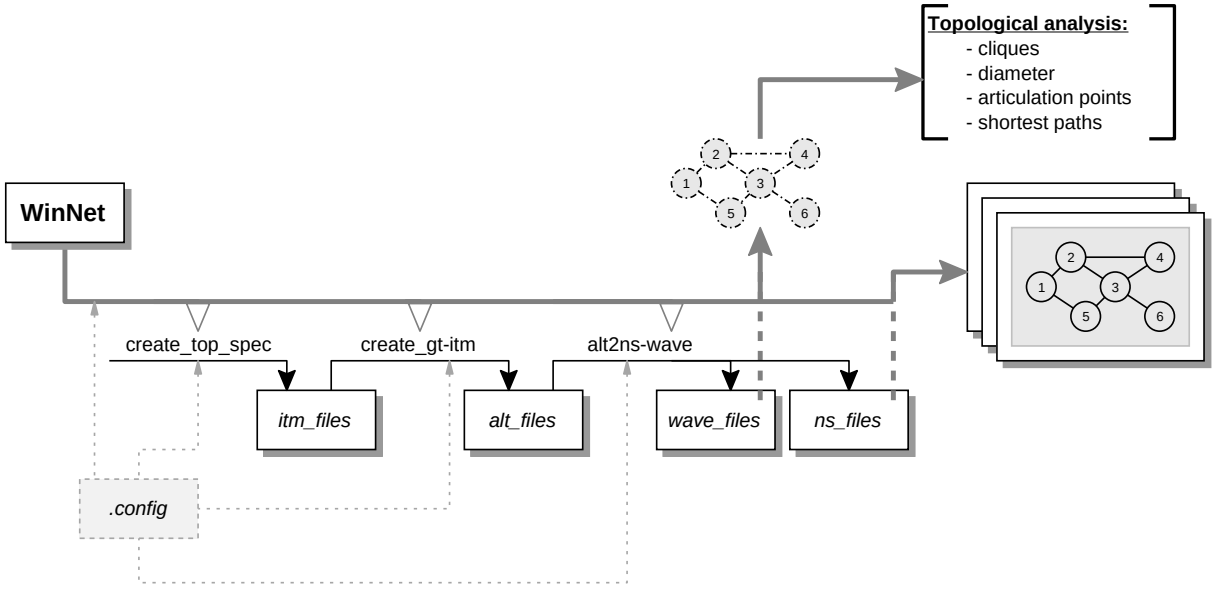


Figure 5: The WinNet flowchart diagram

We used this tool to perform tests on our hierarchical routing protocol called Macro-routing [12] and also to evaluate the performance of the Extended Full-Mesh aggregation [13] that we proposed for finding multiple-constraint hierarchical paths.

[4]Shortest is equivalent, in this context, with BEST in relation with the metric used (i.e. by considering the order and the type of metrics).

3 Summary and Conclusions

We presented in this paper an application called WinNet which uses GT-ITM to generate random network topologies based on which there can be created simulations by using NS and/or virtual network representation, using WAVE, used to analyze the network topology along with the random single or multiple metrics associated with network links. Thus, WinNet can be used:

- as a planning tool by simulating various network configurations and traffic loads.
- as a network analysis tool, to answer questions such as: where will the bottlenecks be in the circuits?; where are the shortest paths?; or what is the diameter and the connectivity on the network?
- as a networking educational tool.
- for solving other problems than the computer network ones (e.g. logic tasks, database searches).

The WinNet application has three main stages which can be accessed separately because they can use files created previously by the application. Moreover, the system can be adjusted to specific needs using a configuration file. Thus, the application can be extended to solve also non-computer network related problems. Moreover, the WAVE representation allows assigning numerical or non-numerical values to nodes as well as to links. This is an important feature for instance, in hierarchical routing, where traversing an aggregated node has an associated cost as well as traversing a link. WinNet's modularity facilitates future changes and extensions for this application.

References

[1] B. Waxman, "Routing of Multipoint Connections," *IEEE Journals on Selected Areas in Communications (JSAC)*, 1988.

[2] K. Calvert and M. Doar and E. Zegura, "Modeling Internet Topology," *IEEE Transactions on Communications*, pp. 160-163, 1997.

[3] M. Doar, "A Better Model for Generating Test Networks," *Proceedings of IEEE GLOBECOM*, November, 1996.

[4] C. Jin and Q. Chen and S. Jamin, "Inet: Internet Topology Generator," *University of Michigan*, Technical Research Report, No. CSE-TR-433-00, 2000.

[5] W. Aiello and F. Chung and L. Lu, "A Random Graph Model for Massive Graphs," *Proceedings of the 32nd Annual Symposium in Theory of Computing*, 2000.

[6] M. Faloutsos and P. Faloutsos and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *ACM Computer Communication Review*, September, 1999.

[7] "Network Simulator - ns - 2," *http://www.isi.edu/nsnam/ns/*, 2002.

[8] András Varga, "OMNeT ++," *IEEE Network Interactive*, Software Tools for Networking, VOL. 16, No. 4, 2002.

[9] Peter S. Sapaty, *Mobile Processing in distributed and Open Environments*, Wiley, ISBN: 0-471-19572-3, 2000.

[10] Peter S. Sapaty, "The WAVE paradigm," Dept. of Informatics, Univ. of Karlsruhe, Germany, No. 17/92, July, 1992, Also published in Proc. Post-Conference Joint Workshop on Distributed and Parallel Implementations of Logic Programming Systems, JICSLP'92, pp. 106-148, Washington, D. C., Nov. 13-14, 1992.

[11] Son T. Vuong and Ivailo Ivanov, "Mobile Intelligent Agent Systems: WAVE vs. JAVA," *Proceedings of the 1st Annual Conference of Emerging Technologies and Applications in Communications (etaCOM'96)*, Portland, Oregon, May, 1996.

[12] Sanda Dragos and Martin Collier, "Macro-routing: a new hierarchical routing protocol," *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, November/December, 2004.

[13] Sanda Dragos and Martin Collier, “The Extended Full-Mesh aggregation technique,” *submitted to ACM Sigcomm 2006*, September, 2006.

Sanda Dragos, Radu Dragos
Babes-Bolyai University
Department of Computer Science
Address: 1, Mihail Kogalniceanu St., Cluj-Napoca, 400084, Romania
E-mail: sanda@cs.ubbcluj.ro, bradu@ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 199-207

Recent Advances in Fuzzy Arithmetics

János Fodor, Barnabás Bede

Abstract: In the present paper we propose a comparative overview of some recent results in fuzzy arithmetics. This study is motivated by the fact that the multiplication of two trapezoidal fuzzy numbers by using Zadeh's extension principle is not of trapezoidal shape. Since in several applications these are the fuzzy numbers which are only admitted, the result of the multiplication is either approximated by a trapezoidal fuzzy number, either the multiplication is given in such a way that the result is of trapezoidal shape. We study several approaches to solve this problem both from the theoretical and practical point of view.

Keywords: fuzzy numbers, fuzzy arithmetics

1 Introduction

Uncertainties in different scientific areas arise mainly from the lack of human knowledge. In many practical situations the uncertainties are not of statistical type. This situation occurs mainly in the case of modeling the linguistic expressions because of their dependence on human judgement. Also, non-statistical uncertainty appears in several situations if we cannot repeat a measurement, so we cannot build a probability distribution of a measured variable. Fuzzy numbers are fuzzy subsets of the set of real numbers satisfying some additional conditions. Fuzzy numbers allow us to model non-probabilistic uncertainties in an easy way.

Arithmetic operations on fuzzy numbers have also been developed, and are based mainly on the extension principle [15] or on interval arithmetic [11]. When operating with fuzzy numbers, the result of our calculations strongly depend on the shape of the membership functions of these numbers. Less regular membership functions lead to more complicated calculations. Moreover, fuzzy numbers with simpler shape of membership functions often have more intuitive and more natural interpretation.

Thus, there is a natural need of simple approximations of fuzzy numbers that are easy to handle and have a natural interpretation. Trapezoidal or triangular fuzzy numbers are most common in current applications.

Considering the extension principle-based arithmetic operations on trapezoidal fuzzy numbers, the product of two such fuzzy numbers is not of the same kind: the shape of these fuzzy numbers is not preserved. This fact leads to serious problems mainly in the case of iterative calculations involving fuzzy numbers. In many situations this problem is solved by approximating the result of the extension principle-based multiplication by a triangular or trapezoidal number. In some other cases the arithmetical operations are defined in a way that the results are always trapezoidal fuzzy numbers.

The aim of the present paper is to give an overview of recent advances in the study of these problems. After the necessary preliminaries we consider two groups of solutions. Methods in the first one try to approximate a fuzzy number with a trapezoidal one. Approaches in the second group re-define the arithmetic operations in such a way that the results are trapezoidal fuzzy numbers directly. At the end we compare the methods and give some conclusion.

2 Preliminaries

Definition 1. A fuzzy number is a function $u : \mathbb{R} \to [0,1]$ with the following properties:

(i) u is normal, i.e., there exists $x_0 \in \mathbb{R}$ such that $u(x_0) = 1$;

(ii) $u(\lambda x + (1-\lambda)y) \geq \min\{u(x), u(y)\}, \forall x, y \in \mathbb{R}, \forall \lambda \in [0,1]$;

(iii) u is upper semicontinuous on $\mathbb{R}$, i.e., $\forall x_0 \in \mathbb{R}$ and $\forall \varepsilon > 0$ there exists a neighborhood $V(x_0)$ such that $u(x) \leq u(x_0) + \varepsilon, \forall x \in V(x_0)$;

(iv) The set $\overline{\text{supp}(u)}$ is compact in $\mathbb{R}$, where $\text{supp}(u) = \{x \in \mathbb{R}; u(x) > 0\}$.

We denote by $\mathbb{R}_{\mathcal{F}}$ the set of all fuzzy numbers.

Let $a, b, c \in \mathbb{R}, a < b < c$. The fuzzy number $u : \mathbb{R} \to [0,1]$ denoted by (a,b,c) and defined by $u(x) = 0$ if $x \leq a$ or $x \geq c, u(x) = \frac{x-a}{b-a}$ if $x \in [a,b]$ and $u(x) = \frac{c-x}{c-b}$ if $x \in [b,c]$ is called a triangular fuzzy number.

For $0 < r \leq 1$ and $u \in \mathbb{R}_{\mathcal{F}}$ we denote $[u]^r = \{x \in \mathbb{R}; u(x) \geq r\}$ and $[u]^0 = \overline{\{x \in \mathbb{R}; u(x) > 0\}}$ the r-level sets (r-cuts) of u. It is well-known that for each $r \in [0,1]$, $[u]^r$ is a bounded closed interval, $[u]^r = [\underline{u}^r, \overline{u}^r]$. Let $u, v \in \mathbb{R}_{\mathcal{F}}$ and $\lambda \in \mathbb{R}$. We define the sum $u+v$ and the scalar multiplication λu by

$$[u+v]^r = [u]^r + [v]^r = [\underline{u}^r + \underline{v}^r, \overline{u}^r + \overline{v}^r]$$

and

$$[\lambda u]^r = \lambda [u]^r = \begin{cases} [\lambda \underline{u}^r, \lambda \overline{u}^r], & \text{if } \lambda \geq 0, \\ [\lambda \overline{u}^r, \lambda \underline{u}^r], & \text{if } \lambda < 0, \end{cases}$$

respectively, for every $r \in [0,1]$.

We denote by $\ominus u = (-1)u \in \mathbb{R}_{\mathcal{F}}$ the negative of $u \in \mathbb{R}_{\mathcal{F}}$.

The product $u \cdot v$ of fuzzy numbers u and v, based on Zadeh's extension principle, is defined by

$$\underline{(u \cdot v)}^r = \min\{\underline{u}^r \underline{v}^r, \underline{u}^r \overline{v}^r, \overline{u}^r \underline{v}^r, \overline{u}^r \overline{v}^r\}$$
$$\overline{(u \cdot v)}^r = \max\{\underline{u}^r \underline{v}^r, \underline{u}^r \overline{v}^r, \overline{u}^r \underline{v}^r, \overline{u}^r \overline{v}^r\}.$$

Surely, the above formulas are not very practical from the computational point of view. Also, let us remark that usually the fuzzy numbers which are used in practical applications are trapezoidal. So, the requirement that a product operation should be shape-preserving seems to be natural.

Definition 2. A fuzzy number $u \in \mathbb{R}_{\mathcal{F}}$ is said to be positive if $\underline{u}^1 \geq 0$, strict positive if $\underline{u}^1 > 0$, negative if $\overline{u}^1 \leq 0$ and strict negative if $\overline{u}^1 < 0$. We say that u and v have the same sign if they are both positive or both negative.

Let $u, v \in \mathbb{R}_{\mathcal{F}}$. We say that $u \prec v$ if $\underline{u}^r \leq \underline{v}^r$ and $\overline{u}^r \leq \overline{v}^r$ for all $r \in [0,1]$. We say that u and v are on the same side of 0 if $u \prec 0$ and $v \prec 0$ or $0 \prec u$ and $0 \prec v$.

Remark 3. If u is positive (negative) then $\ominus u$ is negative (positive).

Definition 4. For arbitrary fuzzy numbers u and v the quantity

$$D(u,v) = \sup_{0 \leq r \leq 1} \{\max\{|\underline{u}^r - \underline{v}^r|, |\overline{u}^r - \overline{v}^r|\}\}$$

is called the (Hausdorff) distance between u and v.

It is well-known (see e.g. [3]) that $(\mathbb{R}_{\mathcal{F}}, D)$ is a complete metric space and D verifies $D(ku, kv) = |k| D(u,v)$, $\forall u, v \in \mathbb{R}_{\mathcal{F}}$, $\forall k \in \mathbb{R}$.

The so-called L-R fuzzy numbers are considered important in fuzzy arithmetics. These and their particular cases triangular and trapezoidal fuzzy numbers are used almost exclusively in applications.

Definition 5. ([4], p. 54, [13]) Let $L, R : [0, +\infty) \to [0,1]$ be two continuous, decreasing functions fulfilling $L(0) = R(0) = 1, L(1) = R(1) = 0$, invertible on $[0,1]$. Moreover, let a^1 be any real number and suppose $\underline{a}, \overline{a}$ be positive numbers. The fuzzy set $u : \mathbb{R} \to [0,1]$ is an L-R fuzzy number if

$$u(t) = \begin{cases} L\left(\frac{a^1 - t}{\underline{a}}\right), & \text{for } t \leq a^1 \\ R\left(\frac{t - a^1}{\overline{a}}\right), & \text{for } t > a^1. \end{cases}$$

Symbolically, we write $u = \left(a^1, \underline{a}, \overline{a}\right)_{L,R}$, where a^1 is called the mean value of u, $\underline{a}, \overline{a}$ are called the left and the right spread. If u is an L-R fuzzy number then (see e. g. [13])

$$[u]^r = \left[a^1 - L^{-1}(r)\,\underline{a}, a^1 + R^{-1}(r)\,\overline{a}\right].$$

As a particular case, one obtains *trapezoidal fuzzy numbers* when the functions L and R are linear. A trapezoidal fuzzy number u can be represented by the quadruple $(a,b,c,d) \in \mathbb{R}^4$, $a \leq b \leq c \leq d$. In this case the r-level sets are given by $\underline{u}^r = a + r(b-a)$ and $\overline{u}^r = d + r(d-c)$. If we have $b = c$ in the representation (a,b,c,d), the fuzzy number is called *triangular*. Then we can use the triple (a,b,d) only.

A trapezoidal fuzzy number (a,b,c,d) can also be represented by a quadruple $\langle m, U, L, R\rangle$, where $m = (b+c)/2$ is the *modal value*, $U = m - b$ is the *upper tolerance*, and $L = m - a$ and $R = d - m$ are the *left and right lower tolerances* respectively (see Figure 1).

The r-level sets of a trapezoidal fuzzy number $u = \langle m_u, U_u, L_u, R_u\rangle$ are given as follows:

$$[u]^r = [m_u - L_u + r(L_u - U_u), m_u + R_u + r(U_u - R_u)]. \tag{1}$$

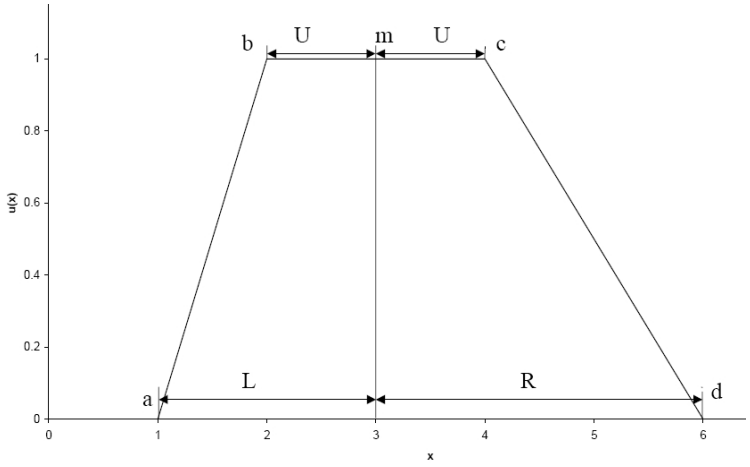


Figure 1: Two representations of a trapezoidal fuzzy number

3 Trapezoidal approximations

3.1 Conventional fuzzy arithmetic with trapezoidal fuzzy numbers

In conventional fuzzy arithmetic with $u, v \in \mathbb{R}_{\mathcal{F}}$, the arithmetic operations $\circ \in \{+, -, \cdot, \div\}$ are defined by applying interval arithmetic to the r-level sets $[u]^r = [\underline{u}^r, \overline{u}^r]$ and $[v]^r = [\underline{v}^r, \overline{v}^r]$ of the fuzzy numbers. The sum and the difference of two trapezoidal fuzzy numbers are also trapezoidal. The product and the quotient are, however, of non-trapezoidal shape.

As an example, consider $u = (0,2,4,6)$ and $v = (2,3,8)$. Then $u+v = (2,5,7,14)$ and $[u \cdot v]^1 = [6,12]$, $[u \cdot v]^0 = [0,48]$, with membership function shown in Figure 2.

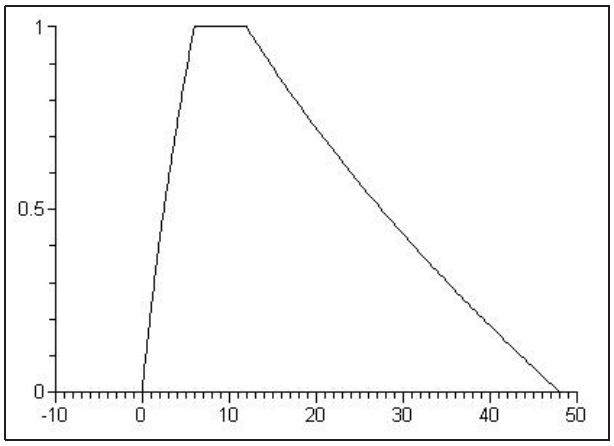


Figure 2: Actual product of $(0,2,4,6)$ and $(2,3,8)$.

Trapezoidal approximation of the conventional product

In applications which use trapezoidal fuzzy numbers, for computational simplicity we often prefer to calculate only the core $[u \cdot v]^1 = [6,12]$ and the support $[u \cdot v]^0 = [0,48]$ using interval methods. The resultant trapezoidal fuzzy number $(0,6,12,48)$ is then used for the approximation of the actual product in Figure 2.

Repeated operations on fuzzy numbers using conventional fuzzy arithmetic may have the effect of increasing the uncertainty with each successive operation. This is inconsistent with the day to day experience and the intuitive way people handle vague quantities [9].

Trapezoidal approximation and least squares fitting

Turning back to the just mentioned trapezoidal approximation of the product of two trapezoidal fuzzy numbers, a very natural idea is to use linear approximations of side functions based on classical least squares. Without going into details, it is interesting to notice that the slopes of these least squares approximations are equal to the respective slopes of the trapezoidal approximation determined by the core and the support of the product. For more details and other results we refer to [5].

3.2 New trapezoidal approximation preserving the expected interval

As we have discussed in the previous sections, the usual (Zadeh's extension principle-based) product does not preserve the shape of the operands. Thus, the result of the product, for computational purposes has to be approximated by trapezoidal number. A new, axiomatic approach has been introduced in [8]. We will call the trapezoidal approximation of the product based on this method as new trapezoidal approximation of the product.

The method proposed in [8] gives the best approximation of the product under some appropiate conditions. These conditions are natural, so in the approximation of the product the result obtained is motivated from the theoretical point of view. Let us regard the trapezoidal approximation as an operator T, $T:\mathbb{R}_{\mathcal{F}}\to\mathbb{R}_{\mathcal{F}}$, which for a given fuzzy number u gives its trapezoidal approximation. The list of requirements which have to be satisfied by an operator of this type are given in [8] below.

a. Conserving some fixed α-cut. E.g. if the operator preserves the 0 and 1 level sets, then the old trapezoidal approximation is reobtained

b. Invariance to translation of the operator T

c. Invariance with respect to rescaling

d. Monotonicity with respect to inclusion

e. Idempotency (i.e. the trapezoidal approximation of a trapezoidal number is itself)

f. To be best approximation, that is, it should be the nearest in some prescribed sence ($D(T(u),u)\leq D(x,u)$ for any trapezoidal number x)

g. Conserves the so-called expected interval, that is the original fuzzy number and its approximation has the same expected interval (Let us recall here that the expected interval of $u\in\mathbb{R}_{\mathcal{F}}$ is $\left[\int_0^1\underline{u}^r dr,\int_0^1\overline{u}^r dr\right]$.

h. Continuity

i. Compatiblity with the extension principle

j. Monotonicity with respect to some ordering between fuzzy numbers

k. Invariance with respect to correlation. (see [7]).

In [8] the authors propose a trapezoidal approximation which is best approximation and preserves the expected interval, that is conditions 6 and 7 are required. In this case, for $u\in\mathbb{R}_{\mathcal{F}}$ we obtain the trapezoidal fuzzy number (t_1,t_2,t_3,t_4), where

$$\begin{aligned}
t_1&=-6\int_0^1 r\underline{u}^r dr+4\int_0^1\underline{u}^r dr,\\
t_2&=6\int_0^1 r\underline{u}^r dr-2\int_0^1\underline{u}^r dr,\\
t_3&=6\int_0^1 r\overline{u}^r dr-2\int_0^1\overline{u}^r dr,\\
t_4&=-6\int_0^1 r\overline{u}^r dr+4\int_0^1\overline{u}^r dr.
\end{aligned}\tag{2}$$

In [8], the authors proved also that conditions 2,3,4,5,8,9,10,11 are fulfiled. Moreover, the expected value of the fuzzy numbers (called also defuzzification by the center of area method) is preserved.

4 New operations resulting in trapezoidal fuzzy numbers

4.1 The cross product

In this section we study the theoretical properties of the cross product of fuzzy numbers. For more details see [1] and [2]. Let $\mathbb{R}_{\mathcal{F}}^{*}=\{u\in\mathbb{R}_{\mathcal{F}}:\ u \text{ is positive or negative}\}$. Firstly we begin with a theorem which was obtained by using the stacking theorem ([12]).

Theorem 6. *If u and v are positive fuzzy numbers then $w=u\odot v$ defined by $[w]^r=[\underline{w}^r,\overline{w}^r]$, where $\underline{w}^r=\underline{u}^r\underline{v}^1+\underline{u}^1\underline{v}^r-\underline{u}^1\underline{v}^1$ and $\overline{w}^r=\overline{u}^r\overline{v}^1+\overline{u}^1\overline{v}^r-\overline{u}^1\overline{v}^1$, for every $r\in[0,1]$, is a positive fuzzy number.*

Corollary 7. *Let u and v be two fuzzy numbers.*

(i) If u is positive and v is negative then $u\odot v=\ominus(u\odot(\ominus v))$ is a negative fuzzy number;

(ii) If u is negative and v is positive then $u\odot v=\ominus((\ominus u)\odot v)$ is a negative fuzzy number;

(iii) If u and v are negative then $u\odot v=(\ominus u)\odot(\ominus v)$ is a positive fuzzy number.

Definition 8. The binary operation $\odot$ on $\mathbb{R}_{\mathcal{F}}^{*}$ introduced by Theorem 6 and Corollary 7 is called cross product of fuzzy numbers.

Remark 9. 1) The cross product is defined for any fuzzy numbers in

$$\mathbb{R}_{\mathcal{F}}^{\wedge}=\{u\in\mathbb{R}_{\mathcal{F}}^{*};\ \text{there exists an unique } x_0\in\mathbb{R} \text{ such that } u(x_0)=1\},$$

implicitly for any triangular fuzzy numbers. In fact, the cross product is defined for any fuzzy number in the sense proposed in [6] (see also [13]).

2) The below formulas of calculus can be easily proved ($r\in[0,1]$):

$$\underline{(u\odot v)}^r=\overline{u}^r\underline{v}^1+\overline{u}^1\underline{v}^r-\overline{u}^1\underline{v}^1,$$
$$\overline{(u\odot v)}^r=\underline{u}^r\overline{v}^1+\underline{u}^1\overline{v}^r-\underline{u}^1\overline{v}^1$$

if u is positive and v is negative,

$$\underline{(u\odot v)}^r=\underline{u}^r\overline{v}^1+\underline{u}^1\overline{v}^r-\underline{u}^1\overline{v}^1,$$
$$\overline{(u\odot v)}^r=\overline{u}^r\underline{v}^1+\overline{u}^1\underline{v}^r-\overline{u}^1\underline{v}^1$$

if u is negative and v is positive. In the last possibility, if u and v are negative then

$$\underline{(u\odot v)}^r=\overline{u}^r\overline{v}^1+\overline{u}^1\overline{v}^r-\overline{u}^1\overline{v}^1,$$
$$\overline{(u\odot v)}^r=\underline{u}^r\underline{v}^1+\underline{u}^1\underline{v}^r-\underline{u}^1\underline{v}^1.$$

3) The cross product extends the scalar multiplication of fuzzy numbers. Indeed, if one of operands is the real number k identified with its characteristic function then $\underline{k}^r=\overline{k}^r=k,\forall r\in[0,1]$ and following the above formulas of calculus we get the result.

The following interpretation related to error theory is a further theoretical motivation of the use of the cross product of fuzzy numbers. Indeed, the consistency of the cross product with the classical theory motivates its use in the case of modeling uncertain data (uncertainty being due to errors of measurement).

We introduce two kinds of errors of fuzzy numbers corresponding to absolute error and relative error in classical error theory and we study these with respect to sum and cross product.

Definition 10. Let u be a fuzzy number. The crisp number $\Delta_L^r(u)=\underline{u}^1-\underline{u}^r$ is called r- error to left of u and the crisp number $\Delta_R^r(u)=\overline{u}^r-\overline{u}^1$ is called r-error to right of u, where $r\in[0,1]$. The sum $\Delta^r(u)=\Delta_L^r(u)+\Delta_R^r(u)$ is called r-error of u.

If u expresses the fuzzy concept A then $\Delta_L^r(u)$ and $\Delta_R^r(u)$ can be interpreted as the values of tolerance of level r from the concept A to left and to right, respectively. For example, if the triangular fuzzy number $u=(5,7,9)$ expresses "early morning" then $\Delta_L^{\frac{1}{2}}(u)=1$ (one hour) is the tolerance of level $\frac{1}{2}$ of u towards night from the concept

of "early morning" and $\Delta_R^{\frac{1}{4}}(u) = 0.5$ (30 minutes) is the tolerance of level $\frac{1}{4}$ of u towards noon from the concept of "early morning".

A new argument in the use of addition of fuzzy numbers as extension (by Zadeh principle) of real addition is the validity of the formula

$$\Delta^r(u+v) = \Delta^r(u) + \Delta^r(v)$$

which is consistent to the classical error theory. It is an immediate consequence of the obvious formulas

$$\Delta_L^r(u+v) = \Delta_L^r(u) + \Delta_L^r(v)$$

and

$$\Delta_R^r(u+v) = \Delta_R^r(u) + \Delta_R^r(v).$$

Now, let us study the relative error of the cross product.

Definition 11. Let u be a fuzzy number such that $\underline{u}^1 \neq 0$ and $\overline{u}^1 \neq 0$. The crisp numbers $\delta_L^r(u) = \frac{\Delta_L^r(u)}{|\underline{u}^1|}$ and $\delta_R^r(u) = \frac{\Delta_R^r(u)}{|\overline{u}^1|}$ are called relative r-errors of u to left and to right. The quantity $\delta^r(u) = \delta_L^r(u) + \delta_R^r(u)$ is called relative r-error of u.

Theorem 12. *If u and v are strict positive or strict negative fuzzy numbers then $\delta^r(u \odot v) = \delta^r(u) + \delta^r(v)$.*

Corollary 13. *If u is a strict positive fuzzy number then $\delta_L^r(u^{\odot n}) = n\delta_L^r(u)$, $\delta_R^r(u^{\odot n}) = n\delta_R^r(u)$ and $\delta^r(u^{\odot n}) = n\delta^r(u)$.*

The above theorems show us that the cross product is consistent with the classical error theory (the propagation of errors is governed by a similar law as in the classical case).

4.2 New arithmetic operations on trapezoidal fuzzy numbers

In [10] a new set of arithmetic operations was introduced for $L-R$ fuzzy numbers. Now we consider its specification for trapezoidal fuzzy numbers. We will employ representation (1) here.

Let $u = \langle m_u, U_u, L_u, R_u \rangle$ and $v = \langle m_v, U_v, L_v, R_v \rangle$ be two trapezoidal fuzzy numbers and consider any arithmetic operation $\circ \in \{+, -, \cdot, \div\}$. By definition from [10],

$$m_{u \circ v} = m_u \circ m_v,$$

and

$$[u \circ v]^r = [m_u \circ m_v - L_r(u,v), m_u \circ m_v + R_r(u,v)]$$

with

$$L_r(u,v) = \max\{L_u - r(L_u - U_u), L_v - r(L_v - U_v)\},$$

$$R_r(u,v) = \max\{R_u - r(R_u - U_u), R_v - r(R_v - U_v)\}.$$

The result is not trapezoidal (see [10]). Its side functions are piecewise linear in general.

Simplified operations on trapezoidal fuzzy numbers

In [14], a simplification of Ma et al's definition was published. Essentially, it is just the traditional trapezoidal approximation applied for Ma et al's result. More formally, let $u = \langle m_u, U_u, L_u, R_u \rangle$ and $v = \langle m_v, U_v, L_v, R_v \rangle$ be two trapezoidal fuzzy numbers and consider any arithmetic operation $\circ \in \{+, -, \cdot, \div\}$. Then define the extension of $\circ$ as follows:

$$u \circ v = \langle m_u \circ m_v, \max\{U_u, U_v\}, \max\{L_u, L_v\}, \max\{R_u, R_v\} \rangle.$$

5 Comparisons

Now we compare from the experimental point of view, four of the approaches outlined above: i) actual product (based on the extension principle), ii) old trapezoidal approximation of the actual product, iii) the cross product, iv) new trapezoidal approximation of the actual product. We do not include in the discussion the method proposed in [10], this being subject of future research.

We have performed several experiments on the above mentioned four operations and we illustrate the results by one sample (the behaviour being the same in all the cases for trapezoidal numbers with positive support). Surely a theoretical validation of the experimental results is necessary and this is subject of future research.

Trapezoidal fuzzy numbers $(1,5,8,10)$, $(1,3,4,8)$ are considered, and the four cases are illustrated in Figure 1. The solid vertical line and the dashed vertical line represent the defuzzified values of the results by centroid and expected values (center of area) methods, respectively.

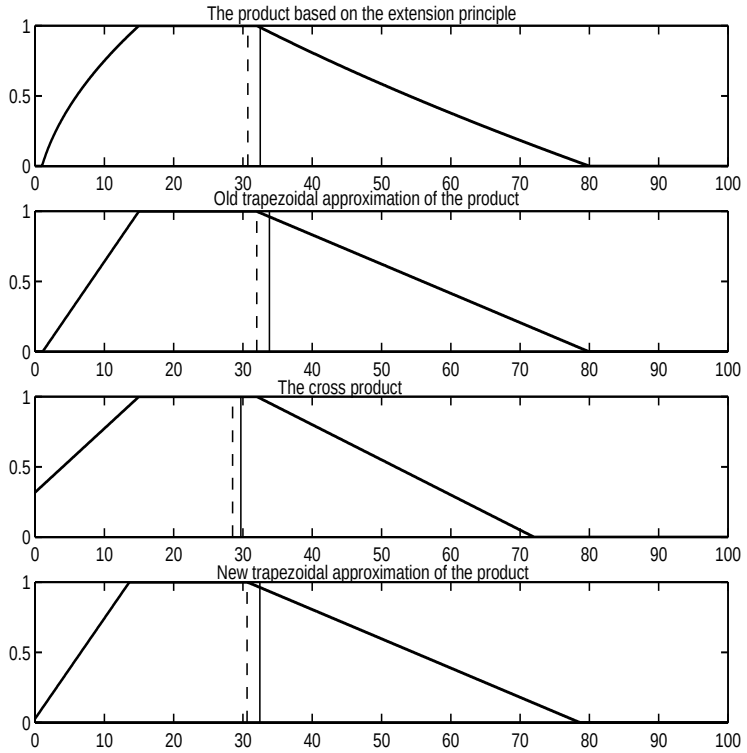


Figure 3: The product of two trapezoidal fuzzy numbers obtained by the different approaches discussed in the paper.

Figure 1 does not show a striking difference between the results of the different methods. However, the difference can be significant if we perform iterative computations with fuzzy numbers. In order to show this we consider the exponential functions obtained as power series with respect to the product type operations discussed above. In Figure 3 the results of the exponential-type functions are presented. The fuzzy number considered in the exponent is $(2.2,4.6,4.7,5)$. Solid thin line represent again defuzzification by centroid method, while dashed line the expected value.

Significant difference can be observed between the different results in this case. Indeed, the iterative use of the product operations leads to different result even after defuzzification. This problem can be avoided by considering and examining all the operations in all the practical problems considered and taking them into account there.

This figure suggests us also that we should be careful with the use of the cross product in the construction of an exponential, since in some cases the result given by this operation is negative, which can never be possible. The figure suggests that probably the best behavior is that of the new trapezoidal approximation.

Also, we observe (from the experimental results) that after defuzzification (by centroid method) the result of the cross product is usually smaller than that of the new trapezoidal approximation, which is smaller at its turn than the old trapezoidal approximation of the product, however the results are not very different. Surely, mathematical

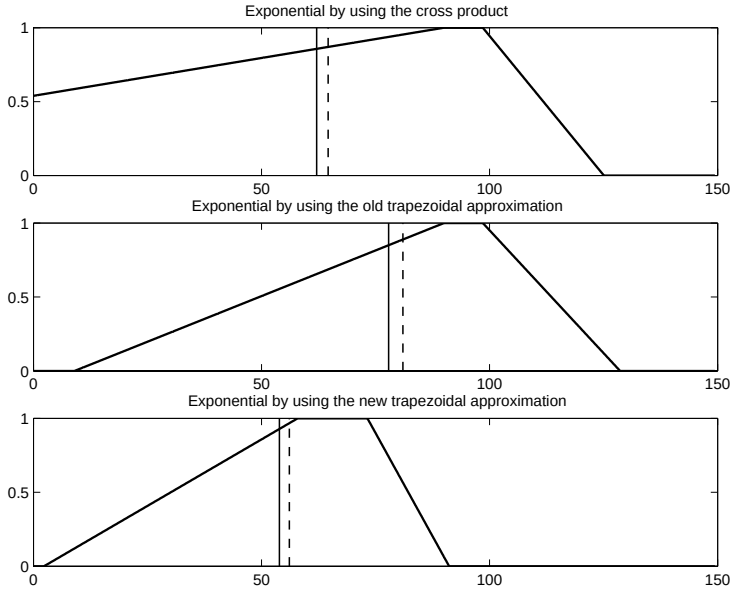


Figure 4: The exponential of a fuzzy number.

investigation of this property is subject of future research.

6 Conclusions and further research

Recent advances in fuzzy arithmetics have been discussed both from theoretical and practical point of view. As a conclusion of this research we can state that the theoretical properties of the cross-product and the new trapezoidal approximation motivate the usefulness of both methods. The most conservative method is the old trapezoidal approximation. So, there exist reasons for using all the above mentioned approaches and to take into account the results of all approaches in applications (e.g., in risk analysis).

References

[1] A.I. Ban, B. Bede, Cross product of fuzzy numbers and properties, *Journal of Fuzzy Mathematics*, to appear.

[2] A.I. Ban, B. Bede, Cross product of $L-R$ fuzzy numbers and properties, *Anals of Oradea Univ. , Fasc. Matem.* 9 (2003), 95-108.

[3] Congxin Wu and Zengtai Gong, On Henstock integral of fuzzy number valued functions (I), *Fuzzy Sets and Systems*, 120 (2001), 523-532.

[4] D. Dubois and H. Prade, *Fuzzy Sets and Systems*, Academic Press, New York, 1980.

[5] J. Fodor, Fuzzy arithmetic and least squares fitting (2006, forthcoming).

[6] R. Fuller and T. Keresztfalvi, On generalization of Nguyen's theorem, *Fuzzy Sets and Systems*, 41 (1991), 371-374.

[7] R. Fullér, P. Majlender, On interactive fuzzy numbers, *Fuzzy Sets and Systems*, 143 (2004), 355-369.

[8] P. Grzegorzewski, E. Mrówka, Trapezoidal approximations of fuzzy numbers, *Fuzzy Sets and Systems*, 153 (2005), 115-135.

[9] G. J. Klir, Fuzzy arithmetic with requisite constraints, *Fuzzy Sets and Systems*, 91 (1997), 165-175.

[10] Ming Ma, M. Friedman and A. Kandel, A new fuzzy arithmetic, *Fuzzy Sets and Systems*, 108 (1999), 83-90.

[11] R. E. Moore, *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

[12] M. L. Puri and D. A. Ralescu, Differentials of fuzzy functions, *J. Math. Anal. Appl.*, 91 (1983), 552-558.

[13] M. Wagenknecht, R. Hampel and V. Schneider, Computational aspects of fuzzy arithmetics based on Archimedean t-norms, *Fuzzy Sets and Systems*, 123 (2001), 49-62.

[14] J. Williams, H. Robinson and N. Steele, Applying Ma et al's new fuzzy arithmetic to triangular and trapezoidal fuzzy numbers, Proc. of the 4th International Conference on Recent Advances in Soft Computing (Nottingham, UK, December 12-13, 2002).

[15] L. A. Zadeh, Fuzzy sets, *Inform. and Control*, 8 (1965), 338-353.

János Fodor, Barnabás Bede
Budapest Tech
Department of Intelligent Engineering Systems
Address: Bécsi út 96/b, H-1034 Budapest, Hungary
E-mail: fodor@bmf.hu, bede.barna@bgk.bmf.hu

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 208-211

Influence of the Parameters in the Learning Algorithm for Travelling Salesman Problem Solved with Kohonen Neural Network

Marieta Gâta, Gavril Toderean

Abstract: In this paper we present an application of Kohonen Neural Network to the Problem of Travelling Salesman. We implemented this algorithm of Travelling Salesman Problem in a Java program. We try to experiment with different values for parameters of algorithm like: number of iteration, learning rate, neighborhood radius, minimum distance, number of neurons, number of cities, etc. and we'll see how they affect the algorithm.

Keywords: Travelling Salesman Problem, Kohonen Neural Networks

1 Introduction

Algorithm

The main idea of Kohonen Neuronal Networks is to leave the network to construct himself. To do this we have to present patterns continuously and randomly until stability is reached. Kohonen Network is composed by two kinds of neurons. First type of neurons has each neuron connected with each neuron (even himself) of this collection. Th weight value of one neuron depends on the distance between neurons. The weight r_{ij} between neuron i and j is given by: $r_{ij} = e^{\frac{-d_{ij}^2}{2*theta^2}}$, where d_{ij} is the Euclidean distance between neuron i and j, and theta is width of the neighborhood (always a positive value). To enter data in the network we need another layer of neurons (the second group of neurons), this layer do not belongs to the same topological rules of the previous network. Each neuron of this group is connected with each neuron of the first group. We will note w the weight that connects the two groups.

The steps of the algorithm are:

a. We situate the neurons of the first group in a solicited final pattern (neurons of first group will have an attribute of position x and y). The final solution is the shorter circle that connects all towns. We will place k neurons, uniformly, in a circle formation, where k is the number of neurons in first group (k $\geq$ number of towns). Input of this network will be the coordinates of the towns. The second group will count 2 neurons, x_i and y_i (n = 2). We initialize all weights with random values. Also we initialize learning rate, indices of iteration t=1, width of the neighborhood.

b. We get a random town and we put his coordinates to input neurons. xi and yi values represent the coordinates of the point.

c. We find the j neuron for that: $d_{ij}^2 = (x_i - w_{xj})^2 + (y_i - w_{yj}^2)$ is minimal. We find the winner neuron using the minimum distance criteria. w_{xj} is the weight value between neuron x_i and neuron j (where j belongs from the first group). Formula is the same for w_{yj}. General formula: $\sum_n (n_i - w_{nj})^2$ is minimal, where n is the n^{th} neuron's input and w_{nj} is the neuron's weight

d. We adjust all the w weights with formula:

$$w_{xi} = w_{xi} + phi * rij * (x_i - w_{xi})$$

$$w_{yi} = w_{yi} + phi * rij * (y_i - w_{yi})$$

for the winner neuron and the neurons which are in the neighborhood, with the learning rate parameter (phi). For general case, the formula looks like:

$$w_{ni} = w_{ni} + phi * rij * (n_i - w_{ni})$$

for all i neurons from the first group and for all n neurons from the second group. This learning is competitive.

e. We decrease with time theta and phi using the following equations: $theta = theta_0 * e^{\frac{-t}{\lambda}}$ and $phi = phi_0 * e^{\frac{-t}{\lambda}}$ where l denotes a time constant, t is the current time-step (iteration of the loop), phi is learning rate, theta is width of the neighborhood function, $theta = theta_0$ and $phi = phi_0$ at the commencement of training (at time t_0). After that we recalculate r_{ij} weights: $r_{ij} = e^{\frac{-d_{ij}^2}{2*theta^2}}$.

f. If phi $<$ epsilon then we go to the step 2 otherwise we stop the algorithm.

2 Observations

After training the weights will be closer to the coordinates of the points. The red path (from Figure1) should be the shortest path to reach all towns. The indices of the points and the relations of the neighborhood will indicate the route.

Learning rate and neighborhood decrease according as the number of iteration is increasing. In phase of ordering, learning rate is about 0.9 and in phase of adjustment learning rate is about 0.02. In step of ordering, the neurons are ordinate so that the neighbors respond to closer class. In this step, learning rate and neighborhood are large. In step of adjusting, the neurons with weights are approach to the prototypes of the classes. These neurons could be the neurons in the neighborhood or even the winner neuron. In this step learning rate and neighborhood are small.

3 Experimental results

We try experimenting with different values for the number of iteration t, initial learning rate phi_0, different neighborhood radius theta, and we'll see how they affect the algorithm.

The learning rate at the start of the training is set to 0.5, after then gradually decays over time so that during the last few iteration it is close to zero.

The width of the neighborhood decreases with number of iteration (time-step), too.

The minimum distances could be Euclidean, Manhattan, and Hamming.

In the next table we try different values for cities and neurons and the program calculates values for phi and theta. Also we want to evaluate the time for different values of epochs. When number of cities remains the same (cities=10 and neurons=20) and number of neurons is rising, the time is risen a little and the degree of 10- remains about the same for phi and theta.

We observe that when the number of neurons remains constant (neurons=10 and cities=20) and the number of cities is increasing, the time remains approximate the same and the degree of 10- remains about the same for the values phi and theta.

Neurons	Cities	Epochs	Time	Phi	Theta
10	10	1000	1'6"	$1.1x10^{-5}$	$1.1x10^{-5}$
10	10	2000	2'7"	$6.1x10^{-10}$	$6.1x10^{-10}$
10	10	3000	3'19"	$3.2x10^{-14}$	$3.2x10^{-14}$

Table 1

Neurons	Cities	Epochs	Time	Phi	Theta
20	10	1000	1'3"	$1.3x10^{-5}$	$1.3x10^{-5}$
20	10	2000	2'4"	$8x10^{-10}$	$8x10^{-10}$
20	10	3000	3'48"	$1.9x10^{-14}$	$1.9x10^{-14}$

Table 2

Neurons	Cities	Epochs	Time	Phi	Theta
10	20	1000	1'6"	$1.1x10^{-5}$	$1.1x10^{-5}$
10	20	2000	2'7"	$6.2x10^{-10}$	$6.2x10^{-10}$
10	20	3000	3'19"	$3.1x10^{-14}$	$3.1x10^{-14}$

Table 3

All this experiments (table 1, table 2, table 3) was effectuated with momentum=0.99 (speed) and near=0.01.

In the next experiments we try different values for moments and the neurons and cities will remains the same (neurons=10, cities=10).

We observe that when the value of momentum decrease (from 0.99 to 0.9 and then to 0.8) the value for phi and theta is decreasing very fast, approximative with the degree of 10^{-1} per second for momentum=0.9 and with the degree of 10^{-2} per second for momentum=0.8. Also we observe that when momentum is decreasing, the time is remains approximative the same. For all this value of momentum (0.8 and 0.9) the accuracy is very coarse.

Momentum (speed)	Near	Epochs	Time	Phi	Theta
0.99	0.01	1000	1'6"	$1.1x10^{-5}$	$1.1x10^{-5}$
0.99	0.01	2000	2'7"	$6.1x10^{-10}$	$6.1x10^{-10}$
0.99	0.01	3000	3'19"	$3.2x10^{-14}$	$3.2x10^{-14}$

Table 4

Momentum (speed)	Near	Epochs	Time	Phi	Theta
0.9	0.01	1000	1'4"	$2.1x10^{-49}$	$2.1x10^{-49}$
0.9	0.01	2000	2'8"	$4.0x10^{-94}$	$4.0x10^{-94}$
0.9	0.01	3000	3'12"	$1.4x10^{-138}$	$1.4x10^{-138}$

Table 5

Momentum (speed)	Near	Epochs	Time	Phi	Theta
0.8	0.01	1000	1'4"	$2.0x10^{-106}$	$2.0x10^{-106}$
0.8	0.01	2000	2'5"	$1.1x10^{-201}$	$1.1x10^{-201}$
0.8	0.01	3000	3'12"	$1.2x10^{-292}$	$1.2x10^{-292}$

Table 6

Initial values are phi=05., theta=0.5, near=0.01, alpha=0.0, minimum distance is Euclidean.

The minimum distances could be Euclidean, Manhattan, and Hamming. If minimum distance is Manhattan: $d_{ij} = (x_i - w_{xj}) + (y_i - w_{yj})$ then these are the current values for this distance:

Neurons	Cities	Epochs	Time	Phi	Theta
10	10	1000	1'4"	$1.8x10^{-5}$	$1.8x10^{-5}$
10	10	2000	2'7"	$7.1x10^{-10}$	$7.1x10^{-10}$
10	10	3000	3'14"	$3.9x10^{-14}$	$3.9x10^{-14}$

Table 7

Neurons	Cities	Epochs	Time	Phi	Theta
20	10	2000	2'38"	$6.4x10^{-10}$	$6.4x10^{-10}$
20	10	3000	3'58"	$3.9x10^{-14}$	$3.9x10^{-14}$

Table 8

Neurons	Cities	Epochs	Time	Phi	Theta
10	20	1000	1'6"	$1.5x10^{-5}$	$1.5x10^{-5}$
10	20	2000	2'11"	$6.0x10^{-10}$	$6.0x10^{-10}$
10	20	3000	3'13"	$3.9x10^{-14}$	$3.9x10^{-14}$

Table 9

Neurons=10, cities=10, near=0.10, momentum=0.99 for values from Table 7, Table 8 and Table 9. We observe that these values are approximative the same as in the case of Euclidean distance.

In Figure 1 we present interface of the program. This program is create in Java J2SDK1.4. Up in the left is a grid with all the neurons and the citis represented. Up in the right are all the values of phi for all neurons. In the corner left-down are all the coordinates for neurons and for the cities. It is calculated dynamic the sum of the path, and it is showed some remarkable valus. In the corner right-down the user can modify some value for studing the influence of these parameters for the algorithm.

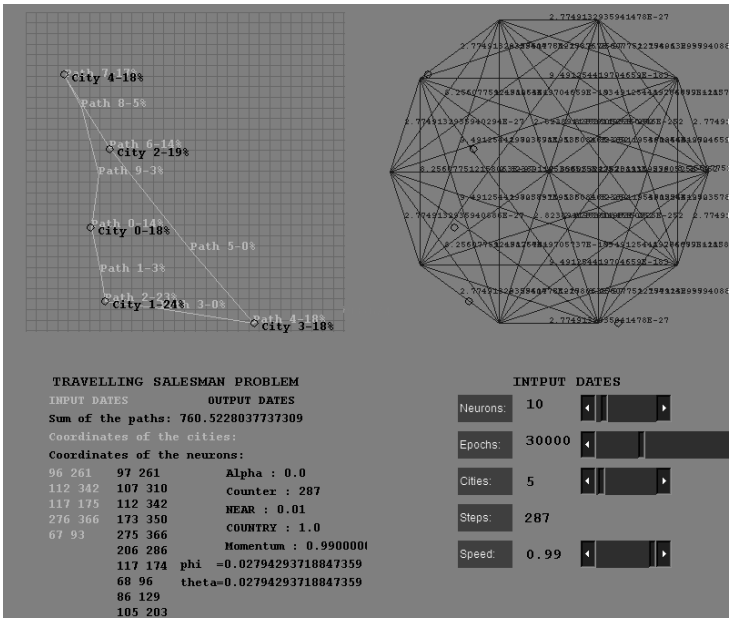


Figure 1: Travelling Salesman Problem resolved with Kohonen Neural Networks

4 Future work

The long term goal is to advance our level of understanding about simulated evolution as a means to configure and optimize Artificial Neural Nets. The medium term goal is to apply our methods to a series of interesting problems. A secondary goal is educational in nature. We attempt to write our software with ample explanation, not just for the user, but for the engineer/programmer who wants to understand the innermost detail.

References

[1] V. B. Rao, "C++ Neural Networks and Fuzzy Logic", *M&T Books, IDG Books Worldwide*, 1995.

[2] L. Patocchi, "Travelling Salesman Problem using Kohonen neuronal network", *http://www.patol.net/java/*.

[3] H. Lohninger, "Learning by simulations", *Teach Me-Data Analysis, Springer*, 1999.

[4] ***, "Traveling Salesman Problem (Princeton)", *http://www.math.princeton.edu/tsp/*.

[5] K. Chen, "Simple learning algorithm for the traveling salesman problem", *Physical Review E*, 1997.

Marieta Gâta
North University of Baia Mare
Department of Mathematics and Computer Science
Address: Victoriei, No. 76, 430144, Baia Mare, Romania
E-mail: marietag@ubm.ro

Gavril Toderean
Technical University of Cluj Napoca
Department of Communications
Address: G. Baritiu, No. 26-28, 400027, Cluj Napoca, Romania
E-mail: Gavril.Toderean@com.utcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 212-215

Rationality of Fuzzy Choice Functions Through Indicators

Irina Georgescu

Abstract: In this paper we study the rationality of a fuzzy choice function C by means of the rationality indicators $Rat_G(C)$ and $Rat_M(C)$. $Rat_G(C)$ expresses the degree of G-rationality of C and $Rat_M(C)$ expresses the degree of M-rationality of C.

Keywords: fuzzy choice function, rationality indicators, multiple criteria decision–making

1 Introduction

Samuelson's theory of revealed preference [11] expresses the rationality of a consumer in terms of some preference relation associated with a demand function. Uzawa [15], Arrow [1], Richter [10] , Sen [12] and many others enlarged Samuelson's theory of revealed preference introducing in an abstract setting the concept of choice function.

Real world cases have required the assumption of vague preferences and vague choices. Vague preferences are modelled by fuzzy preference relations [4]. Even if the preference is ambiguous, the choice can be exact or vague. When the choice is exact it will be described by a crisp choice function (see [9] for a detailed discussion).

There exist cases (negotiations on electronic marketplaces) when the decision maker cannot make a definitive choice. In this process of decision making, the choice is potential [2]. Consequently we need to consider fuzzy choice functions. Papers [5], [6], [7] develop a theory of revealed preference for a large class of choice functions. The choice functions studied in these papers include these of Banerjee [2].

In this paper there are defined two indicators: the indicator of G-rationality $Rat_G(C)$ and the indicator of M-rationality $Rat_M(C)$. $Rat_G(C)$ expresses the degree to which the choice function C is G-rational and $Rat_M(C)$ expresses the degree to which C is M-rational. Besides these rationality indicators, we introduce two indicators of normality studied $Norm_G(C)$ and $Norm_M(C)$.

The indicators $Rat_G(C)$ and $Rat_M(C)$ refine the notions of G–rationality and M–rationality and $Norm_G(C)$ and $Norm_M(C)$ refine the notions of G–normality and M–normality.

These indicators allow us to compare the fuzzy choice functions from their rationality point of view. Accordingly, for two fuzzy choice functions C_1 and C_2, when $Rat_G(C_1) \leq Rat_G(C_2)$ we can consider that C_2 is superior to C_1 with respect to G–rationality.

The four indicators are connected by the relation $Norm_M(C) \leq Rat_M(C) \leq Rat_G(C) = Norm_G(C)$.

Another result is the inequality $Norm_G(C) \leq F\alpha(C)$ where $F\alpha(C)$ is an indicator that expresses the degree to which consistency condition $F\alpha$ is verified [7].

We prove the equality $\mathcal{A}(C) = Norm_G(C) \wedge Trans(R)$ where $\mathcal{A}(C)$ is the Arrow indicator of C [8] and $Trans(R)$ is the degree of transitivity of the revealed preference relation R associated with C.

Instead of deciding whether or not a choice function is rational, it is better to measure the degree to which a choice function is rational. These indicators of rationality allow us to compare the choice functions from their rationality point of view.

There are cases when a multiple criteria decision making problem can be converted into a fuzzy choice problem. In order to establish the corresponding fuzzy choice problem, the existing information can be subject to the analysis of several independent experts. The result of each expert leads to a fuzzy choice problem; we will have as many choice problem proposals as many experts are. The decision-maker will select the one that reflects more faithfully the reality. He will make his decision according to a principle of rationality: the best decision will correspond to the fuzzy choice problem with the highest degree of rationality.

Our intention is to prove the efficaciousness of such concepts and results by using them in multiple criteria decision making problems.

2 Preliminaries

For any $\{a_i\}_{i\in I} \subseteq [0,1]$ we denote $\bigvee_{i\in I} a_i = \sup\{a_i | i \in I\}$ and $\bigwedge_{i\in I} a_i = \inf\{a_i | i \in I\}$. We consider the residuum $\rightarrow$ of the Gödel t-norm $\wedge$ $a \rightarrow b = \bigvee\{c \in [0,1] | a \wedge c \leq B\}$ and the biresiduum $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$ [3].

Let X be a non-empty set. A fuzzy set A of X is a function $A : X \to [0,1]$. We denote by $\mathcal{F}(X)$ the family of fuzzy subsets of X. $A \in \mathcal{F}(X)$ is normal if $A(x) = 1$ for some $x \in X$. If $x_1, \ldots, x_n \in X$ then $[x_1, \ldots, x_n]$ will be the characteristic function of $\{x_1, \ldots, x_n\}$. For $A, B \in \mathcal{F}(X)$ we denote

$I(A,B) = \bigwedge_{x \in X} (A(x) \to B(x))$; $E(A,B) = \bigwedge_{x \in X} (A(x) \leftrightarrow B(x))$.

$I(A,B)$ is called the subsethood degree of A in B and $E(A,B)$ is called the degree of equality of A and B.

A fuzzy preference relation on X is a function $Q : X^2 \to [0,1]$. Q is said to be

• transitive if $Q(x,y) \wedge Q(y,z) \leq Q(x,z)$ for any $x,y,z \in X$;

• strongly complete if $Q(x,y) = 1$ or $Q(y,x) = 1$ for any $x,y \in X$.

For any fuzzy preference relation Q on X we define

$Trans(Q) = \bigwedge_{x,y,z \in X} [(R(x,y) \wedge R(y,z)) \to R(x,z)]$

$SC(Q) = \bigwedge_{x,y \in X} [Q(x,y) \vee Q(y,x)]$.

It is obvious that $Trans(Q) = 1$ iff Q is transitive and $SC(Q) = 1$ iff Q is strongly complete. $Trans(Q)$ will be called the degree of transitivity of Q and $SC(Q)$ the degree of strong completeness of Q.

3 Rational Fuzzy Choice Functions

A fuzzy choice space is a pair $\langle X, \mathcal{B}$ where X is a non-empty universe of alternatives and $\mathcal{B}$ is a non-empty family of non-zero fuzzy subsets of X. Following [5], [6], a fuzzy choice function on $\langle X, \mathcal{B}\rangle$ is a function $C : \mathcal{B} \to \mathcal{F}(X)$ such that for any $S \in \mathcal{B}$, $C(S)$ is a non-zero fuzzy subset of X and $C(S) \subseteq S$. For any $S \in \mathcal{B}$ and $x \in X$, $S(x)$ is the availability degree of alternative x. In interpretation, every available set $S \in \mathcal{B}$ corresponds to some criterion or attribute of alternatives. Our definition of a fuzzy choice function generalizes Banerjee's [2]. In [2] the domain of a fuzzy choice function is made of all non-empty finite subsets of X and the range is made of fuzzy subsets of X. In our approach, both the domain and the range of a choice function contain fuzzy subsets of X.

Let $\langle X, \mathcal{B}\rangle$ be a fuzzy choice space and Q a fuzzy preference relation on X. For any $S \in \mathcal{B}$ let us define the fuzzy subsets $G(S,Q)$ and $M(S,Q)$ of X.

$G(S,Q)(x) = S(x) \wedge \bigwedge_{y \in X} [S(y) \to Q(x,y)]$

$M(S,Q)(x) = S(x) \wedge \bigwedge_{y \in X} [(S(y) \wedge Q(y,x)) \to Q(x,y)]$

In the crisp case $G(S,Q)$ is the set of Q-greatest elements of X and $M(S,Q)$ is the set of Q-maximal elements of X. It is easy to see that $G(S,Q) \subseteq M(S,Q)$. In general $G(.,Q)$ and $M(.,Q)$ are not fuzzy choice functions. If Q is reflexive and transitive and X is finite then $G(.,Q)$ and $M(.,Q)$ are fuzzy choice functions.

A fuzzy choice function C is G-rational (resp. M-rational) if $C = G(.,Q)$ (resp. $C = M(.,Q)$) for some fuzzy preference relation Q on X. To a fuzzy choice functions one assigns the revealed preference relation R on X defined by

$R(x,y) = \bigvee_{S \in \mathcal{B}} (C(S)(x) \wedge S(y))$ for any $x,y \in X$.

C is called G-normal (resp. M-normal) if $C = G(.,R)$ (resp. $C = M(.,R)$).

Let C_1, C_2 be two fuzzy choice functions on $\langle X, \mathcal{B}\rangle$. We define the degree of similarity $E(C_1,C_2)$ of C_1 and C_2 by $E(C_1,C_2) = \bigwedge_{S \in \mathcal{B}} \bigwedge_{x \in X} (C_1(S)(x) \leftrightarrow C_2(S)(x))$.

4 Rationality Indicators

Let $\langle X, \mathcal{B}\rangle$ be a fuzzy choice space and C a fuzzy choice function on $\langle X, \mathcal{B}\rangle$. Denote by $\mathcal{R}$ the set of fuzzy preference relations on X. Define

$Rat_G(C) = \bigvee_{Q \in \mathcal{R}} E(C, G(.,Q))$, $Norm_G(C) = E(C, G(.,R))$,

$Rat_M(C) = \bigvee_{Q \in \mathcal{R}} E(C, M(.,Q))$, $Norm_M(C) = E(C, M(.,R))$.

For any set X, $Norm_G(C) = 1$ iff C is G-normal and $Norm_M(C) = 1$ iff C is M-normal.

The number $Rat_G(C)$ (resp. $Rat_M(C)$) is called the indicator of G-rationality (resp. M-rationality) of C. $Rat_G(C)$ (resp. $Rat_M(C)$) expresses the degree to which the choice function C is G-rational (resp. M-rational). $Norm_G(C)$ (resp. $Norm_M(C)$) is called the indicator of G-normality (resp. M-normality) of C and evaluates the degree to which C is G-normal (resp. M-normal).

Theorem 1. $Norm_M(C) \leq Rat_M(C) \leq Rat_G(C) = Norm_G(C)$.

Theorem 2. *If Q is a fuzzy preference relation on X then $SC(Q) \leq E(G(.,Q),M(.,Q))$.*

For a fuzzy choice function C we define
$ScRat(C) = \bigvee_{Q\in\mathcal{R}} (E(C,G(.,Q)) \wedge SC(Q))$

Theorem 3. $ScRat(C) = \bigvee_{Q\in\mathcal{R}} (E(C,M(.,Q)) \wedge SC(Q))$.

$ScRat(C)$ is called the indicator of strongly complete rationality of C.

Theorem 4. $Norm_M(C) \leq ScRat(C)$.

Following [6], we say that C verifies the fuzzy consistency condition $F\alpha$ if for all $S,T \in \mathcal{B}$ and $x \in X$, $I(S,T) \wedge S(x) \wedge C(T)(x) \leq C(S)(x)$. $F\alpha$ is a fuzzy version of Sen's consistency condition α [12]. For a fuzzy choice function C we define
$F\alpha(C) = \bigwedge_{S,T\in\mathcal{B}} \bigwedge_{x\in X} [I(S,T) \wedge S(x) \wedge C(T)(x)) \rightarrow C(S)(x)]$.
It is easy to see that $F\alpha(C) = 1$ iff C verifies $F\alpha$.

Theorem 5. $Norm_G(C) \leq F\alpha(C)$.

Cf. [8], C verifies the Fuzzy Arrow Axiom (*FAA*) if for any $S_1,S_2 \in \mathcal{B}$ and $x \in X$,
$I(S_1,S_2) \wedge S_1(x) \wedge C(S_2)(x) \leq E(S_1 \cap C(S_2), C(S_1))$.
Let us define the Arrow index of a fuzzy choice function C by
$\mathcal{A}(C) = \bigwedge_{S_1,S_2\in\mathcal{B}} \bigwedge_{x\in X} [(I(S_1,S_2) \wedge S_1(x) \wedge C(S_2)(x)) \rightarrow E(S_1 \cap C(S_2), C(S_1))]$.
It is obvious that $\mathcal{A}(C) = 1$ iff C satisfies *FAA*.
Recall hypotheses $H1$ and $H2$ [5]:
$H1$ Every $S \in \mathcal{B}$ and $C(S)$ are normal fuzzy subsets of X;
$H2$ $\mathcal{B}$ includes the fuzzy sets $[x_1,\ldots,x_n]$ for any $n \geq 1$ and $x_1,\ldots,x_n \in X$.

Theorem 6. *If C satisfies hypotheses $H1$ and $H2$ then $\mathcal{A}(C) = Norm_G(C) \wedge Trans(R)$.*

References

[1] K.J. Arrow, "Rational Choice Functions and Orderings," *Economica*, Vol. 26, pp. 121–127, 1959.

[2] A. Banerjee, "Fuzzy Choice Functions, Revealed Preference and Rationality," *Fuzzy Sets and Systems*, Vol. 70, pp. 31–43, 1995.

[3] R. Bělohlávek, *Fuzzy Relational Systems. Foundations and Principles*, Kluwer, 2002.

[4] J. Fodor, M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Kluwer, Dordrecht, 1994.

[5] I. Georgescu, "On the Axioms of Revealed Preference in Fuzzy Consumer Theory," *Journal of Systems Science and Systems Engineering*, Vol. 13, pp. 279–296, 2004.

[6] I. Georgescu, "Consistency Conditions in Fuzzy Consumers Theory," *Fundamenta Informaticae*, Vol. 61, pp. 223–245, 2004.

[7] I. Georgescu, "Revealed Preference, Congruence and Rationality: A Fuzzy Approach," *Fundamenta Informaticae*, Vol. 65, pp. 307–328, 2005.

[8] I. Georgescu, "Arrow's Axiom and Full Rationality for Fuzzy Choice Functions," *Social Choice and Welfare*, forthcoming.

[9] P. Kulshreshtha, B. Shekar, "Interrelationship Among Fuzzy Preference - Based Choice Function and Significance of Rationality Conditions: A Taxonomic and Intuitive Perspective," *Fuzzy Sets and Systems*, Vol. 109, pp. 429–445, 2000.

[10] M. Richter, "Revealed Preference Theory," *Econometrica*, Vol. 34, pp. 635–645, 1966.

[11] P.A. Samuelson, "A Note of the Pure Theory of Consumer's Behavior," *Economica*, Vol. 5, pp. 61–71, 1938.

[12] A.K. Sen, "Choice Functions and Revealed Preference," *Review of Economic Studies*, Vol. 38, pp. 307–312, 1971.

[13] K. Suzumura, "Rational Choice and Revealed Preference," *Review of Economic Studies*, Vol. 43, pp. 149–159, 1976.

[14] K. Suzumura, *Rational Choice, Collective Decisions and Social Welfare*, Cambridge University Press, Cambridge, 1983.

[15] H. Uzawa, "A Note on Preference and Axioms of Choice," *Annals of the Institute of Statistical Mathematics*, Vol. 8, pp. 35–40, 1956.

Irina Georgescu
Academy of Economic Studies, Bucharest, Romania
Department of Economic Cybernetics
Address: Calea Dorobanţilor 15–17, P. O. Box 1–432, 014700, Bucharest
E-mail: irina.georgescu@csie.ase.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 216-221

A Portal Application for Accessing Grid Resources and Services

Alexandru Gherega, Felicia Ionescu

Abstract: Grid computing technologies offers specific mechanisms to allow heterogeneous resource and data sharing over wide spread network areas with improved QoS forming a single virtual system image by providing transparency of the mechanisms involved for the end-user. The Grid systems entities are assured with interoperability aspects through use of a standard interface which translates user requirements and resource specifications into a common language regardless of hardware, software and operating system of the resource. In this settings the resource and data coordination and monitoring become an important aspect due to considerable diversity and dynamic behavior of entities with which a user might want to interact. We present here a portal application architecture by which non-grid users can access Grid resources. The main components of our architecture are the resource management and monitoring Grid services: MiddlewareService and StatusService. The Grid system and services were developed using the Globus Toolkit open source software.

Keywords: computational grid, grid computing, grid service, portal.

1 Introduction

In the early 90's researchers in computing science area were exploring the design and development of a new distributed computing technology and infrastructure which could allow geographically dispersed resources to be aggregated into a single virtual system which in turn provides reliable, secure and easy access to computing power. An analogy emerged by looking at the power grids model and evolution which provided reliable, low-cost access to standardized services with the result that power became universally available. Thus the computational Grid term emerged and is used to define the new infrastructure that is looking, in its development, to enable the increase in computing power by managing unused or underutilized distributed resources in a reliable and secure manner. A Grid system integrates this infrastructure by allowing sharing and accessing of a multitude of heterogeneous, geographically dispersed computing resources including computing clusters, supercomputers, data storage systems and dedicated computing equipments.

The most used definitions for Grid computing are those given by Ian Foster and Carl Kesselman, two of the pioneers of computational Grids, and the Globus Alliance.

A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.
(Carl Kesselman and Ian Foster)[1][2]

According to the Globus Alliance (globus.org) grids are persistent environments that enable software applications to integrate instruments, displays, computational and information resources that are managed by diverse organizations in widespread locations.

The main difference between grid computing and classic distributed computing is the special attention that is given to coordinating the resources shared by entities which present a dynamic life-cycle. Thus beside basic concepts provided by distributed computing Grid technology provides flexible job distribution based on diverse paradigms (peer-to-peer, client/server, etc.), precise control levels over resource utilization, authentication and authentication credentials delegation.[3]

The basic architectural model used in Grids is the Simple Object Architecture SOA through which architectural layers are described using the abstract model of services. A service is defined as an independent entity which provides a specific functionality, is provided with inter-services communication capabilities and it is defined in high correlation with elements that form this service interaction protocol. Based on SOA an application is formed from a collaborative collection of such entities generically called services.

Harnessing the computing power provided by Grid environments is no easy task. Thus non-grid users (i.e users which dispose of local resources but are not included in the Grid environment) must adhere to the specific Grid environment use-case policies. One way of simplifying the process is through use of Grid portals. A Grid portal is an application server enabled with the necessary Grid services and resource interaction mechanisms and user interfaces which allows them to use the Grid systems resources.

2 Grid Architecture

The main goal of Grid technology architecture is to ensure interoperability and the appropriate resource distribution mechanisms between any possible participant. In this perspective the interoperability aspect pursues the definition of corresponding protocols which enables resource negotiation and the distributed environment management and exploitation.

The interoperability aimed by the Grid architecture addresses the possibility of initializing the distributed environment between the Grid participants and the dynamical integration of new members while providing system, platform, programming environment and programming language independence. Without achieving such an interoperability level over organizations boundaries, policies and heterogeneous resources the participants would be restrained to a bilateral agreement interaction policy which limits the necessary dynamism level of Grid systems.

With respect to such contexts sets of rules (i.e protocols) were required to define:

- a standard way to assure system distributed entities interaction to obtain the specific Grid behavior;
- a structure for the interchanged information during entities interaction.

Resource discovery mechanism, identity evaluation and authentication mechanisms and interaction initiation mechanism in distributed environments must be provided and must be flexible enough to allow easy development and future extensions. Thus the Grid architecture provides mechanisms that do not require significant changes in local used policies while allowing highest level of priority in managing one's own resources preservation.

As mentioned in the introductory part of this paper the Grid architecture is service orientated by adopting the SOA model[4]. A service is an abstraction to define entities which provide specific functionalities (i.e a flow of operations to solve a specific problem) and is defined in close correlation with the protocol used to invoke and interact with this service in a network environment. Defining and using standard services as the architecture metric allows service sharing and resource specific details hiding in a distributed environment such as the Grid.

When developing a Grid system two services categories are enabled: low-level services and high-level services. Low-level services category provides resource management services, monitoring and discovery services, information services, interrogation services, security services. High-level services category is formed by the application's specific services. Most challenging issues in Grid enabled environments are those addressing resource management, access and planning mechanisms.[4]

3 Portal application for accessing a Grid environment

The application logic focuses on providing the functionality of a portal[1] and offering a secure, distributed access to computing resources using Grid services. The developed Grid services are implemented using the Java programming language and are deployed using under the container provided by the Globus Toolkit.

Application Architecture

A three-tier architecture was used for the portal application design. The three levels are formed from the Grid resources, application server and the application client.

The portal application assures consistent and secure access to a series of resources represented by applications developed using Grid services. For didactic reasons the chosen applications were the PI and FFT computations. The PI computation is solved through use of an approximation method which involves the integration of the

[1] *Portal* Application or service that offers a broad array of resources and services to geographically dispersed users.

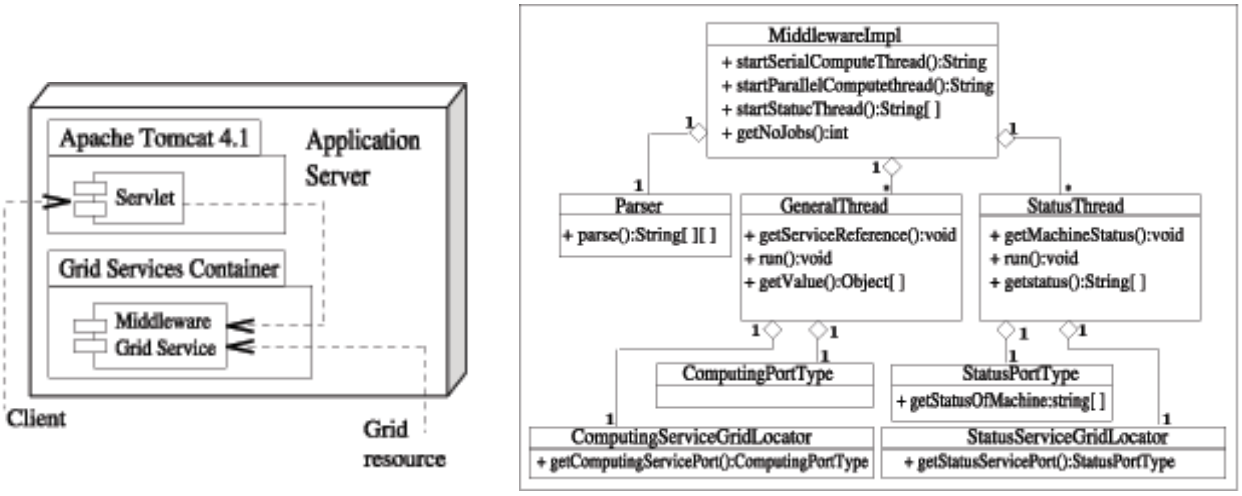
$4/(1+x^2)$ curve in the [0;1] interval. The FFT computation is implemented using the Cooley-Tukey algorithm. The main difference between these two algorithms is that the PI algorithm can be divided in independent tasks while the FFT can not (i.e the FFT decomposition forms tasks that are dependent one on other). Services access and through them resource access is provided in a transparent manner to the client. To obtain user level transparency regarding Grid services access the application server involves concurrent use and interaction of two servers with different tasks: the user interface server and resource and Grid services management server. The server which manages the user interface runs the following operations: ges the user input data, communication of the structured input data to the management Grid service (second server), supplies the answer given by the management server to clients. The user interface is made up of Java servlet software components deployed under Apache Tomcat 4.1 container.

The resource management and monitoring server is implemented through a Grid service called MiddlewareService which is deployed on a Grid container supplied by Globus Toolkit software.

MiddlewareService Grid Service Design

The MiddlwareService design pursues the following main tasks: application job monitoring and management. The monitoring of jobs is solved by the MiddlewareService by invoking a Grid service deployed on each machine in the Grid environment, called StatusService. Using this service the MiddlewareService gets the resource specific informations such as processor frequency, total memory, free/occupied memory at a given moment. Beside this task the service counts the number of grid applications jobs that executes on each machine in the Grid.

Another role given to the MiddlewareService, beside job monitoring and management, is to call the computing Grid services deployed on the Grid machines (workers). In case the application invoked by the user presents a parallel work flow the MiddlewareService distributes the jobs so that a concurrent execution is obtain. In the case that the application has a serial work flow the service sends the hole task (without dividing it into jobs) to a specified machine (user âĂŞ choice).



(a) (b)

Figure 1: (a) Application Server (b) MiddlewareService class diagram

The MiddlewareService is designed as a persistent entity since its functionality requires:

- system monitoring which involves a life cycle which spans the life cycle of all of its client;
- given the information it provides are general in nature whit respect to the given resource and not to an application the life span of the services must overrun the life span of any application which executes in the Grid.

Persistence adds a bonus to the design: less memory consumption (i.e one service instance is used for each client). The Grid resources management server uses three additional Java classes that defines objects used by the MiddlewareService:

- the StatusThread.class designed and used to define a thread object which invokes the StatusService Grid service;
- the GeneralThread.class designed and used to define a thread object which calls the worker Grid services;
- the Pasres.class designed and used to interpret a text file that contains names and URLs of each machine in the Grid System. This file contains lines with the following structure:
  machine_name , http://machine's_ip : port/ogsa/services/gridservices/core.
  The main advantage and reason for such a file is to add a plus of dynamism(i.e the registration of a new machine in the Grid system with respect to the portal becomes a simple editing task).

StatusService Grid Service Design

This Gird service is called by the MiddlewareService in order to obtain the following resource related informations:

- resource (worker machine) processor frequency;
- resource total memory;
- resource free/occupied memory at a given time;
- resource OS name, version and architecture.

The StatusService provides to the outer world one method: getStatusOfMachine() (i.e the service's methods exposed through its interface). This method returns a vector of String Java objects containing the information described above. Thus the StatusService offers a monitoring functionality over the resource memory. The other information are sampled only once and store for further use. Given that this service acts as monitor its life cycle must overlast the clients life cycle. Thus the StatusService is also design as a persistent Grid service.

Application deployment

Application deployment is a two stage process: servlet components which make up the user interface are deployed under the Tomcat Apache 4.1 container; Grid services deployment under the Globus Toolkit container. This process is described in the deployment diagram in *figure 2* which presents the server machine and one of the workers machines.

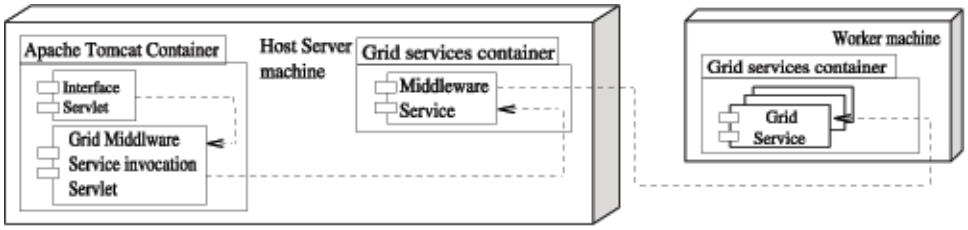


Figure 2: Portal application deployment diagram

On every machine part of the Grid systems Grid services containers are installed with respect to the Certification Authority CA - which resides on the host server machine (i.e all machines use digital certificates signed by the host CA). For the CA the standard simple CA provided by the Globus Toolkit has been used. On the host server machine are deployed the MiddlewareService and the corresponding servlet components which make up the user interface. On all other machines (i.e not the host server machine) called workers are deployed the computing services and the StatusService.

The application's work flow is described in the sequence diagrams below. Figure 3 illustrates the sequence diagram between the application client, the servlet components and the MiddlwareService. Thus the client invokes the servlet method service(..) which takes two arguments. Further on the servlet calls the MiddlwareService which invokes the StatusService on the corresponding machine. After getting the machine state information the client

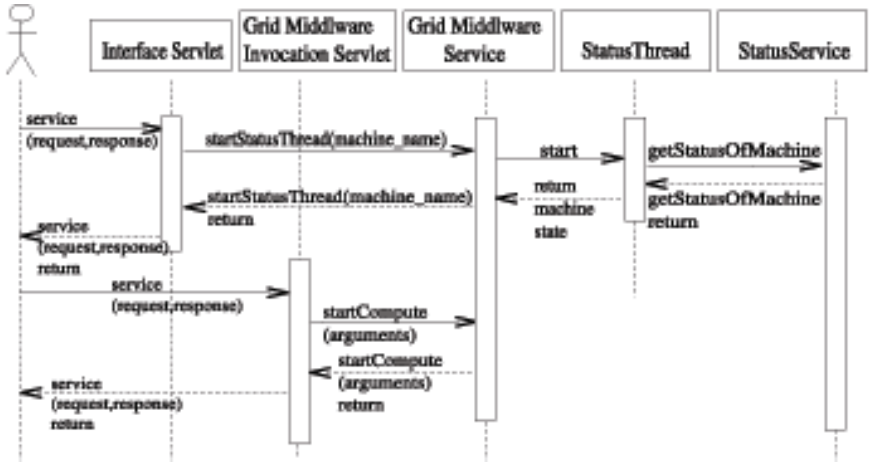


Figure 3: Client, servlet components and MiddlewareService interactions sequence diagram

proceeds with asking for some computation. Figure 4 presents the sequence diagram for a parallel flow application managed by the MiddlewareService.

A two step interface is provided to the user. First a list of available applications is provided to the user.

After a choice has been issued a second interface is provided. At this level the GridMiddleware invocation servlet calls on the MiddlewareService for informations about the resources and number of jobs. The MiddlewareService calls the StatusService to obtain specific resource infromations and returns them to the client. At this point the user can choose on which resource(s) to execute the selected application. If a serial application is involved only one resource can be selected. If a parallel flow is provided than the MiddlewareService divides the main task into smaller sub-tasks and submits them to the selected resources (int this case multiple resources can be selected). The application logic for this case is presented in figure 4.

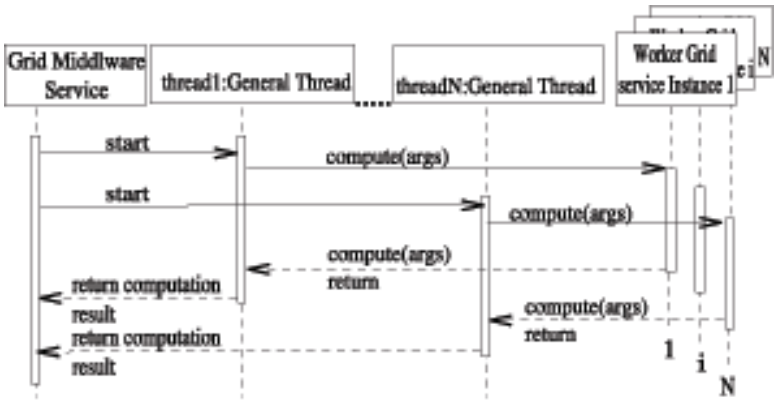


Figure 4: Parallel execution flow application managed by the MiddlewareService sequence diagram

If a parallel application is provided than based on the input data and the number of Grid resources available for solving the corresponding computation the main task is divided by the MiddlewareService into a number of N subtasks which have no kind of interdependencies. The Middlewareservice defines a number of threads equal with the number of subtasks. Each thread will than call on the worker Grid service on different machines an provides it with the input data necessary to solve it's allocated subtask. Being on different resources (i.e execution of each subtask takes place on a different processor) the subtasks can be executed concurrently - if there are no interdependencies. After all subtasks have finished the MiddlewareService collects and assembles the final result and returns to the client.

While the MiddlewareService and StatusService which form the application server are designed as persistent services the worker services are designed as non-persistent. The main reason for such a design approach is because

a single service instance for a very large number of users would mean a substantial fall in execution performance (i.e speed). It would also mean additionally mechanisms that would refer to computation synchronization âĂŞ if a number of users would access concurrently the same worker service (without any delay or queue mechanism at the MiddlewareService level) than scrambled results may appear.

4 Conclusions

While Grid computing environments provides higher performance levels within its boundaries than other distributed technologies, expanding access policies to outside users can be a daunting task. The Grid portal application solves part of this complex problem by providing an architecture through which ordinary users can benefit from Grid environments capabilities.

Thus a non-grid user can still be provided with access to resources and application by using the portal. The main focus area of the application architecture lays within designing Grid services that provide reliable resource management and monitoring.

While the portal can be used for the sole purpose of offering some specific application to users as with classical Web applications the portal may serve as a starting point in user's resource authentication, registration and integration within a Grid environment thus allowing other resources to be added to the system based on a mutual agreement between the user and the Grid environment administration authority.

Such a growing system provided with the necessary services for resource allocation and management and services for negotiating security policies between the variety of distinct administrative domains may be the new generation of Internet as predicted by Ian Foster.

References

[1] I. Foster, C. Kesselman, "The Grid:Blueprint for a New Computing Infrastructure," *Morgan-Kaufmann*, July 1998

[2] I. Foster, C.Kesselman, "The Anatomy of The Grid," *IJSA*, 2001

[3] L. Wolfgang, J. I. Vestgarden,B. Nurdlund, "Grid and related technologies," *Norwegian Computing Center âĂŞ Applied Research and Technologies*, 16th June 2004

[4] L. Ferreira, A. Thakore, M. Brown, "Grid Services programming and Application Enablement," *www.ibm.com/redbooks, IBM*, May 2004

[5] G. Zaglakas, "Grid Computing: Meeting the challenges of an On Demand world," *The Rational Edge*, 15th September 2004

Alexandru Gherega, Felicia Ionescu
Politehnica University of Bucharest, Romania
Faculty of Electronics, Telecommunications and Information Technology
Address: 21 Pipera St., Bucharest
E-mail: alex.gherega@gmail.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 222-227

Mealy Membrane Automata: An Automata-like Approach of Membrane Computing

Mihai Gontineac

Abstract: The purpose of this paper is to give a model for the dynamical aspects of membrane computing.The dynamic flavor is given in terms of direct products of Mealy Multiset Automata and resource device distributors.

Keywords: membrane computing, P-systems, Mealy multiset automata, Mealy membrane automata

1 Introduction

In this paper we present a first step toward a Mealy version of a membrane automaton corresponding to membrane computing defined by P systems [16], i.e. a Mealy membrane automaton that corresponds to an single membrane, then we present the way to connect it with other Mealy membrane automata from its neighborhood

Membrane systems represent a new abstract model of parallel and distributed computing inspired by cell compartments and molecular membranes [16]. A cell is divided in various compartments, each compartment with a different task, and all of them working simultaneously to accomplish a more general task of the whole system. The membranes of a P system determine regions where objects and evolution rules can be placed. The objects evolve according to the rules associated with each region, and the regions cooperate in order to maintain the proper behaviour of the whole system. Membrane systems provide a nice abstraction for parallel systems, and a suitable framework for distributed and parallel algorithms [6]. It does not exist yet a programming language based on, or inspired of, the membrane systems. Thinking to the programming point of view, a sequential software simulator of membrane systems is presented in [10], and a parallel simulator implemented on a cluster of computers is presented in [9]. However it is desirable to find more connections with various fields of computer science, including the classic automata theory. There exist some previous attempts [13, 11, 15, 3, 1]: [13] and [15] present membrane automata as P automata, devices that works mainly with communication rules; [11] presents a P transducer as a form of Mealy membrane automata but it is not so obvious the dynamical aspect; in [3] there is a preliminary study of the dynamics of P systems, which is not based on automata - anyhow, the open problems list at the end of this paper makes our attempt quite natural; in [1] there is defined the concept of EC P automata as devices that accepts/rejects strings but there is not defined any transition function, the notion that deals with dynamic aspects. We have to mention here the existence of devices that also accepts multisets, i.e. multiset automata, introduced in [12].

We introduce in [7] two versions of Mealy - like automata (Mealy multiset automata (MmA) and the elementary Mealy membrane automata (eMMA)) and we continue the study of the (co)algebraic properties of MmA in [8].

We define here an improved version of the elementary Mealy Membrane Automaton namely *simple Mealy Membrane Automaton* (sMMA) by extending the communication capabilities, then we provide an example on how we can use it to obtain a dynamical model of a membrane system.

2 Some Algebra of Multisets

We denote by $\mathbb{N}\langle A\rangle$ the set of all multisets on A, i.e. $\{\alpha : A \to \mathbb{N} \mid \alpha \text{ is a mapping}\}$ (we are inspired for this notation from the notion of group ring, that has the same approach). The structure of $\mathbb{N}\langle A\rangle$ is mainly an additive one, since we add multiplicities of appearance.

This argument is sustained also by the chemical reactions that are the base of the biological modelling. They provide a notation for defining the way a biological system evolves. A chemical reaction has a generic form like it follows:

$$aA + bB \xrightarrow{k} cC + dD.$$

Generally speaking, in algebra the sum is commutative, while the multiplication is not, so this is another fact that makes the above mentioned operation with multisets to be of additive nature.

If $\alpha, \beta \in \mathbb{N}\langle A\rangle$ their *sum* is the multiset $(\alpha+\beta) : A \to \mathbb{N}$ defined by $(\alpha+\beta)(a_i) = \alpha(a_i) + \beta(a_i), i = \overline{1,n}$. Moreover, if we consider the letters from A as multisets, i.e. $a_i := \mu_{a_i} : A \to \mathbb{N}$, where $\mu_{a_i}(a_i) = 1$ and $\mu_{a_i}(a_j) = 0$

for all $j \neq i$, we can express every multiset $\alpha \in N\langle A\rangle$ as a linear combination of a_i, $\alpha = \sum_{i=1}^{n} \alpha(a_i) \cdot a_i$. We can also define an *external operation*, $m\alpha = \sum_{i=1}^{n} (m\alpha(a_i)) \cdot a_i$, for all $m \in \mathbb{N}$ and for all $\alpha \in \mathbb{N}\langle A\rangle$, providing $\mathbb{N}\langle A\rangle$ with a structure of $\mathbb{N}$-semimodule (semimodule over the semiring of positive integers).

We shall also need the *difference* between two multisets over $A(A^*)$, defined by $(\alpha - \beta)(w) = \alpha(w) - \beta(w)$ for all α, β such that $\alpha \supseteq \beta$ (i.e. $\alpha(w) \geq \beta(w)$ for all w).

3 Mealy Multiset Automata (MmA)

3.1 The Definitions

In order to give a suitable model of the molecular functions of a membrane, we need the notion of *Mealy multiset automata* (MmA). Roughly speaking, a MmA consists of a *storage location* (a *box* for short) in which we place a multiset over an input alphabet and a device to translate the multiset into a multiset over an output alphabet. The way in which the MmA works is given by: we have a detection head that detects whether or not a given multiset appears in the multiset available in the box. If the multiset is detected, then it is removed from the box and the automaton inserts a multiset over the output alphabet (or a marked symbol if the output alphabet is the same) that can not be viewed by the detection head. Our automaton stops when no further move is possible. We say that the submultiset read by the head was translated to a multiset over the output alphabet. From the formal point of view,

Definition 1. A *Mealy multiset automaton* is a construct

$$\mathcal{A} = (Q, V, O, f, g, q_0)$$

where,

Q is a finite set, the set of *states*;
$q_0 \in Q$ is special state, which is both initial and final;
V is a finite set of objects, the *input alphabet*;
O is a finite set of objects, the *output alphabet*, such that $O \cap V = \emptyset$;
$f : Q \times \mathbb{N}\langle V\rangle \to \mathcal{P}(Q)$is the *state-transition* (*partial*) *mapping*;
$g : Q \times \mathbb{N}\langle V\rangle \to \mathcal{P}(\mathbb{N}\langle O\rangle)$ is the *output* (*partial*) *mapping.*
If $\mid f(q,a) \mid \leq 1$ we say that $\mathcal{A}$ is *Q- deterministic* and if $\mid g(q,a) \mid \leq 1$ our automaton is *O-deterministic.*

Till now, we have the same definition as for the classical Mealy automata. A MmA is also endowed with a box where it receives a multiset. After that, it begins to process this multiset over V passing through different *configurations.* It starts with a multiset from $\mathbb{N}\langle V\rangle$ and ends with a multiset from $\mathbb{N}\langle V \cup O\rangle$.

To be more specific:

Definition 2. A *configuration* of $\mathcal{A}$ is a triple $(q, \alpha, \overline{\beta})$ where $q \in Q, \alpha \in \mathbb{N}\langle V\rangle$, $\overline{\beta} \in \mathbb{N}\langle O\rangle$. We say that a configuration (q, α, β) *passes* to $(s, \alpha - a, \beta + \overline{b})$ (or, that we have a *transition* between those configurations) if there is $a \subseteq \alpha$ such that $s \in f(q,a), \overline{b} \in g(q,a)$. We denote this by $(q, \alpha, \beta) \vdash (s, \alpha - a, \beta + \overline{b})$. We also denote by $\vdash^*$ the reflexive and transitive closure of $\vdash$.

Remark 3. We could alternatively define a configuration to be a pair (q, α) where $\alpha \in \mathbb{N}\langle V \cup O\rangle$ and the transition relation is $(q, \alpha) \vdash (s, \alpha - a + \overline{b})$, with the same conditions as above.

Definition 4. A multiset $\alpha \in \mathbb{N}\langle V\rangle$ is said to be a *totally consumed multiset* (*tc-multiset*) for $\mathcal{A}$ if, starting from the configuration $(q_0, \alpha, \varepsilon)$ the MmA can pass through configurations till it arrives in a configuration $(q_0, \varepsilon, \beta)$ (i.e. there exists $(q_0, \alpha, \varepsilon) \vdash^* (q_0, \varepsilon, \beta)$).

A multiset $\alpha \in \mathbb{N}\langle V\rangle$ is said to be a *consumed multiset* (*c-multiset*) for $\mathcal{A}$ if, starting from the configuration $(q_0, \alpha, \varepsilon)$ the MmA can pass through configurations till it arrives in a configuration (q, ε, β) (i.e. there exists $(q_0, \alpha, \varepsilon) \vdash^* (q, \varepsilon, \beta)$).

In those cases, we say also that α was *entirely translated to* $\overline{\beta}$.

In all the other situations we say that $\alpha \in \mathbb{N}\langle V\rangle$ is *partially consumed* (*pc-multiset*), or it is *partially translated.*

Notation. Denote by $TC(\mathcal{A})$ the set of all tc-multisets of $\mathcal{A}$, by $C(\mathcal{A})$ the set of all c-multisets of $\mathcal{A}$, and by $PC(\mathcal{A})$ the set of all pc-multisets of $\mathcal{A}$.

Remark 5. The set of c-multisets of the automaton is also of interest for us, since those kind of multisets can be translated. It is not so interesting for one that needs only the accepting behaviour of the automaton.

Theorem 6. *$TC(\mathcal{A})$ is a $\mathbb{N}$- subsemimodule of $\mathbb{N}\langle V\rangle$. Moreover, if we put $\mathcal{A}(\alpha)=\bar{\beta}$ for all $\alpha\in TC(\mathcal{A})$, we obtain an $\mathbb{N}$-homomorphism from $TC(\mathcal{A})$ to $\mathbb{N}\langle O\rangle$.*

Remark 7. It is possible for two multisets $\alpha,\alpha'\in\mathbb{N}\langle V\rangle$ to have their sum in $TC(\mathcal{A})$,even they are not in $TC(\mathcal{A})$.

$C(\mathcal{A})$ does not have the same property, i.e. in general it is not a $\mathbb{N}$- subsemimodule of $N\langle V\rangle$. Let us consider $\alpha,\alpha'\in C(\mathcal{A})$. We have $(q_0,\alpha,\varepsilon)\vdash^*(q,\varepsilon,\beta)$ and $(q_0,\alpha',\varepsilon)\vdash^*(q',\varepsilon,\beta')$, so $(q_0,\alpha+\alpha',\varepsilon)\vdash^*(q,\alpha',\bar{\beta})$ and it is possible that the automaton can not go further (it is possible, for example, that $f(q,a')=\emptyset,(\forall)a'\subseteq\alpha'$).

For some of the categorical properties of MmA's, as well as *behaviour* and *bisimulation relation*, we refer to [7],[8]

3.2 Restricted direct product of Mealy multiset automata

Let $\mathcal{A}_i=(Q_i,V,O,f_i,g_i)$, and B_i their corresponding boxes, $i=\overline{1,n}$, a finite family of Mealy multiset automata. We can connect them in *parallel* in order to obtain a new MmA defined by $\mathcal{A}=\bigwedge_{i=1}^{n}\mathcal{A}_i=(\times_{i=1}^{n}Q_i,V,O,f,g)$, called the *restricted direct product* of $\mathcal{A}_i$, where:

- $f((q_1,q_2,...,q_n),a)=(f_1(q_1,a),f_2(q_2,a),...,f_n(q_n,a))$
- $g((q_1,q_2,...,q_n),a)=(g_1(q_1,a),g_2(q_2,a),...,g_n(q_n,a))$
- The box of $\mathcal{A},B$, will be the disjoint union of $\{B_i\mid i=\overline{1,n}\},\bigsqcup_{i=1}^{n}B_i$.
- A *configuration* of $\mathcal{A}$ is a triple $(q,\alpha,\bar{\beta})$, where $q=(q_1,q_2,...,q_n),\alpha=(\alpha_1,\alpha_2,...,\alpha_n),\bar{\beta}=(\bar{\beta}_1,\bar{\beta}_2,...,\bar{\beta}_n)$
- The (*asynchronous*) *transition relation* of $\mathcal{A}$: $(q,\alpha,\bar{\beta})\vdash(s,\alpha-a,\bar{\beta}+\bar{b})$ iff there is at least an $i\in\overline{1,n}$ such that $s_i\in f_i(q_i,a_i),\bar{b}_i\in g_i(q_i,a_i)$.

We choose the asynchronous way of transition due to the biological nature: "The biological systems are massively concurrent, heterogeneous, and asynchronous" [4].

4 Mealy Membrane Automata

While, as we already mentioned in the introduction, almost all models are based on "top-down" communication rules, and the "maximal parallel and non-deterministic" way of application of the rules is more or less visible, we want to present a new approach, based on automata like systems, such that every membrane is able to evolve and communicate. We are speaking also about the *maximal parallel and non-deterministic* approach. While the parallel part is given by the way we connect various MmA's, the "maximality" and the "non-deterministically" parts will be insure by a "smart" device that is formalized by a *resource mapping* (RM). The advantages of such an approach come from the fact that we proceed from the bottom, i.e. we start from very simple membranes (more or less elementary), then we indicate the way to connect them such that those devices can be used for P systems, as well as for tissue-like P systems.

4.1 Simple Mealy membrane automata (sMMA)

Generally speaking an *simple Mealy Membrane Automata* is built from:

- A *resource box B* together with a *resource mapping*, *RM*;
- *n* Mealy multiset automata, connected in parallel. They consume and translate tc-multisets from their boxes, allocated by *RM*, and the translation results in marked multisets over the same alphabet. We use marked multisets for the output because the input alphabet and the output one must be disjoint. The marking shows us where the corresponding multiset must go (we shall define also the "neighborhood")

Attached to the machine we have an *distribution map* denoted by *DM*. The *DM* is involved in refreshing the content of the box and in communication with other MMA. It is also useful for defining a kind of cascade product of the MMA with itself, allowing us to speak about computations (see , also, the figure).

>From the formal point of view:

Definition 8. A *simple Mealy membrane automaton* (shortly, sMMA) is a machine $\mathcal{A} = (\bigwedge_{i=1}^{n} \mathcal{A}_i, RM, B)$, where:

- $\mathcal{A}_i = (Q_i, V, V \times \{tar\}, f_i, g_i), tar \in \{in_j\}$ are MmA connected in parallel and j belongs to an indexed set of sMMA
- $RM : \mathbb{N}\langle V\rangle \to \mathcal{P}(\mathbb{N}\langle V\rangle^{n+1})$ is the *resource mapping*, where $RM(w) \in \{(k_1x_1, k_2x_2, ..., k_nx_n, w') \mid k_1x_1 + k_2x_2 + ... + k_nx_n + w' = w, x_i$ is a tc-multiset of $\mathcal{A}_i$, and $x_i \nsubseteq w', i = \overline{1,n}\}$
- B is the box where $\mathcal{A}$ receive a multiset for processing.

We have to explain some features of our automaton. Every $\mathcal{A}_i$ has the possibility to process not only one tc-multiset. This corresponds to the possibility of changing the rules after every step of a computation in the P systems case.

The non-deterministic appliance of rules from P systems is given here by the way *RM* is defined, i.e. using "$\in$"and not "$=$". The maximality is given also by *RM*, since the multiset that remains unprocessed in the box is minimal (does not contain any submultiset that can be processed by a MmA from the direct product).

Of course, we can also set some priorities in the way *RM* makes the distribution of resources, but in this case we loose the maximal parallel and non-deterministic feature.

We can express a priority of, say, $\mathcal{A}_1$ over $\mathcal{A}_2$forcing k_2 to be 0 any time we have $k_1 \neq 0$. See the example that should clarify this.

Another thing that makes sMMA a versatile notion is the fact that we do not make any reference to hierarchical systems, so we can use the notion for P-systems as well as for tissue-like P-systems.

4.2 Connecting sMMA

We can define the *neighborhood* of a sMMA $\mathcal{A}$ to be all the simple Mealy Membrane Automata that can communicate with $\mathcal{A}$. In a P-system environment (i.e. hierarchical system of sMMA's) by a neighborhood of $\mathcal{A}$ we understand its parent, its children and itself.

The output of a sMMA $\mathcal{A} = (\bigwedge_{i=1}^{n} \mathcal{A}_i, RM)$ is a multiset from $\mathbb{N}\langle V \times \{tar\}\rangle$ of the form $(w'', here) + \sum_{i=1}^{n} k_i \cdot (y_i, tar_i)$ so it can be viewed as a translation mapping from $\mathbb{N}\langle V\rangle$ to $\mathbb{N}\langle V \times \{tar\}\rangle$ (w'' is the updated content of the box; it contains the unprocessed multiset w' added with possible other multisets from its neighborhood). In order to proceed to the next step of translation, we shall need to show the way to connect the sMMA with other sMMA's. This can be done using a cascade-like product using the *distribution mapping* $DM : \mathbb{N}\langle V \times \{tar\}\rangle \to (\mathbb{N}\langle V\rangle)^m$, where m is the number of sMMA's in the neighborhood of $\mathcal{A}$.

$$DM(\sum_{i=1}^{m} k_i \cdot (y_i, tar_i)) = (w_1 + k_1 \cdot y_1, w_2 + k_2 \cdot y_2, ..., w_m + k_m \cdot y_m),$$

where w_i represents the box's content of the sMMA indexed by i.

A graphical representation of such a network is the following:

This network has a skin-like membrane (the sMMA indexed with 0), two children (those indexed with 1 and 2) and the sMMA indexed with 3 which is the child of the sMMA indexed with 1. As far as we can observe, number 3 can communicate only with 1, number 1 can communicate only with 0 and 3, and 2 can communicate only with 0 and so on.

If we denote by $N(i)$ the set of indexes of the sMMA's in the neighborhood of the sMMA denoted by i, in our diagram we can emphasize the following sets: $N(0) = \{0,1,2\}, N(1) = \{0,1,3\}, N(2) = \{0,2\}, N(3) = \{1,3\}$.

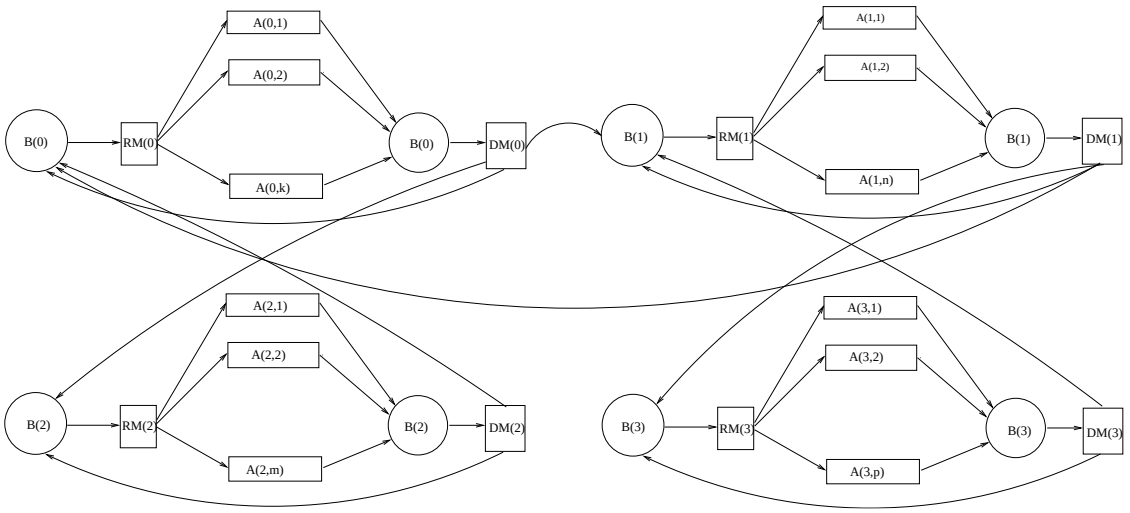
We give now an example of P-system and we describe a network of sMMA's that do the same job.

Example 9. *Consider the P-system with two membranes*

$\Pi = (\{a,b,c,d,e,f\}, \{e\}, \emptyset, [_1[_2]_2]_1, na + mb + c, \emptyset, (R_1, \rho_1), (\emptyset, \emptyset), 2)$

where:

- $R_1 = \{a + c \to f, r_1 : f + b \to f + d + (e,2), r_2 : f \to c, r_3 : d \to b\}$



- $\rho_1 = \{r_1 > r_2, r_1 > r_3\}$

As it can be easily seen, this P-system computes *nm*.

We consider now two sMMA's, $\mathcal{A}^1 = (\bigwedge_{i=1}^{4} \mathcal{A}_i^1, RM^1, B^1)$ and $\mathcal{A}^2 = (\emptyset, \emptyset, B^2)$.

$\mathcal{A}_1^1$ "knows " to translate $a+c$ in $(f,1)$, $\mathcal{A}_2^1$ translates $f+b$ in $(f,1)+(d,1)+(e,2)$, $\mathcal{A}_3^1$ translates f in $(c,1)$, and, finally, $\mathcal{A}_4^1$ translates d in $(b,1)$.

RM^1 is defined as

$RM^1(k_1a+k_2b+k_3c+k_4d+k_5e+k_6f) = (l_1(a+c), l_2(f+b), l_3f, l_4d, w')$, where

$l_1 = min\{k_1,k_2\}, l_2 = min\{k_2,k_6\}, l_3 = \begin{cases} 0 & l_2 \neq 0 \\ k_6 & l_2 = 0 \end{cases}, l_4 = \begin{cases} 0 & l_2 \neq 0 \\ k_4 & l_2 = 0 \end{cases}$

The distribution mapping is defined by

$DM(l_1(f,1)+l_2((f,1)+(d,1)+(e,2))+l_3(d,1)+l_4(b,1)) = (w'+(l_1+l_2)f+(l_2+l_3)d+l_4b, w"+l_2e)$, where w' and $w"$ are the contents of B^1, B^2, respectively, before applying distribution.

It can be verified that if we insert initially in the box $na+mb+c$, after doing all the computations, we receive $c+md$ in B^1 and mne in B^2.

5 Conclusions and further research

Automata theory has mainly a sequential nature, in contrast with P-systems. Membrane systems represent abstract models inspired by the compartments of a cell. We try to connect the theory of membrane computing with the classic theory of (Mealy) automata.

On the other hand, our machine has the capability to be highly adaptable, i.e. we can easy pass from strings to multisets and back, we can also deal with the accepting task, and so on. The inductive description is not able to distinguish between deterministic and non-deterministic automata. As we are more interested in their behaviour, we shall present a co-inductive point of view. It is necessary to point out some interesting facts: while strings are of algebraic nature, multisets can be also viewed as their duals, so they have a coalgebraic nature.

References

[1] A. Alhazov. Minimizing Evolution-Communication P Systems and EC P Automata.

[2] O. Andrei, G. Ciobanu, D. Lucanu. Executable Specifications of the P Systems. In *Membrane Computing WMC5*, Lecture Notes in Computer Science vol.3365, Springer, 127-146, 2005.

[3] F. Bernardini, V. Manca. Dynamical aspects of P systems. Biosystems, 70:85-93, 2002

[4] L.Cardelli. Languages and notations for systems biology. *Unconventional Programming Paradigms*, Invited talk, Le Mount St.Michel, 2004.

[5] D. Besozzi, G. Ciobanu. A P System Description of the Sodium-Potassium Pump. In G.Mauri, Gh.Păun, M.J.Perez-Jimenez, G.Rosenberg, A.Salomaa (Eds.):*Membrane Computing WMC5*, Lecture Notes in Computer Science 3365, Springer, 211-2234, 2005.

[6] G. Ciobanu. Distributed algorithms over communicating membrane systems. *BioSystems* vol.70(2), 123-133, Elsevier, 2003.

[7] G. Ciobanu, M. Gontineac. Mealy Multiset Automata, IJFCS 17 (no.1), 111-126, 2006

[8] G. Ciobanu, V.M. Gontineac. Algebraic and Coalgebraic Aspects of Membrane Computing, Lecture Notes in Computer Science, vol. 3850, Springer, 181-198, 2006

[9] G. Ciobanu, W. Guo. P Systems Running on a Cluster of Computers. In *Membrane Computing*, Lecture Notes in Computer Science vol.2933, Springer, 123-139, 2004.

[10] G. Ciobanu, D. Paraschiv. P System Software Simulator. *Fundamenta Informaticae* vol.49, 61-66, 2002.

[11] G. Ciobanu, Gh. Păun, Gh. Ştefănescu. Sevilla Carpets Associated with P Systems, *Tech. Report 26*, Rovira i Virgili University, 135-140, 2003. To appear in *New Generation Computing*, Springer, 2005.

[12] E. Csuhaj-Varju, C. Martin-Vide, V. Mitrana. Multiset Automata. In C.Calude,Gh. Păun, G.Rozenberg, A.Salomaa (Eds.): *Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View*, LNCS 2235, Springer, 69-83, 2001.

[13] E. Csuhaj-Varju, G. Vaszil. P automata or purely communicating accepting P systems, in Gh. Păun, G. Rozenberg, A. Salomaa, C. Zandron (Eds.): *Membrane Computing. WMC-CdeA 2002*, LNCS 2597, Springer, Berlin, 219–233, 2003.

[14] M.Holcombe. *Algebraic Automata Theory*, Cambridge University Press, 1982.

[15] M. Oswald. *P Automata*, PhD Thesis, Faculty of Computer Science, TU Vienna, 2004.

[16] Gh. Păun. *Computing with Membranes: An Introduction*, Springer, 2002.

Mihai Gontineac
"A. I. Cuza" University
Department of Mathematics
Address: Bd. Carol I, no.11, Iasi, Romania
E-mail: gonti@uaic.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 228-232

Interstructure - A Concept for Add New Generation of Telecommunication Technologies in Transportation Field

Florin Domnel Grafu

Abstract: Based on communication transition toward much more performance for physical media and a better stability from telecommunication device standardization point of view involved, we shown that the new generation of telecommunication equipments and new management models from transportation area could be merged in **interstructure**. Which brings new this concept **interstructure** is that it separate transportation telecommunication area from infrastructure and superstructure and involve this one in a single term so that make easier to distinguish in practice. In other words, when we discuss about **interstructure** maintenance we exclude from the outset that we operate on the infrastructure component that is placed in a specific area or on the superstructure existing in that particular site. We tackle this subject from the transportation point of view, because in this domain there are specific studies and the differences between infrastructure and superstructure are clearly delimited.

Keywords: *interstructure*, Ethernet, communication, transportation.

1 Introduction

We can affirm that is the first time when this subject of **interstructure** is discussed and this word doesn't exist at this moment in any dictionary. There are no other studies about this type of approach, so the research will start with this study. If we start with the necessity of integrated communication infrastructures and add the needs of management services, we will see that these services become very important for developing new ideas and add new possibilities to improve public services in transportation area. This new concept introduces the separation of all software and hardware equipments from what we call infrastructure or superstructure of a specific area of interest for population. We will try to define **interstructure** concept and refer primarily at transportation telecommunications then we will show the utility of concept.

2 The definition of a new concept for transportation telecommunication

The new concept is defined as:

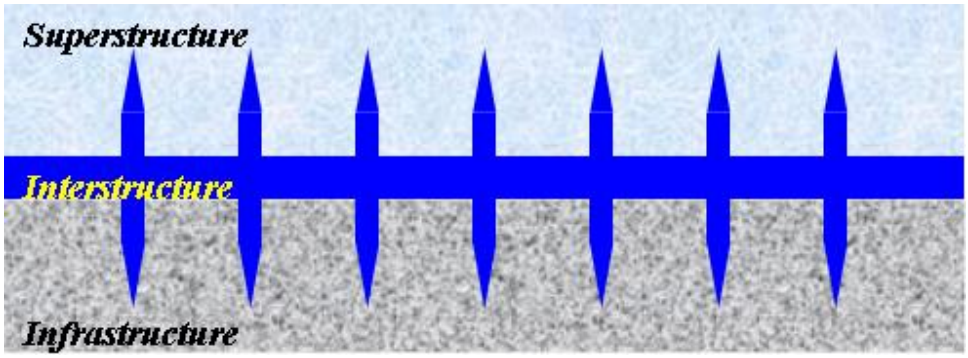


Figure 1: ***Interstructure*** affiliation zone- conceptual representation

Interstructure
n.
1. All components that assure the concomitant, directly or indirectly maintenance for a construction and for any

entity that use it.
2. All equipments that assure a good functionality for infrastructure and superstructure through interworking of maintenance elements.
So, this concept is the entity which ties the infrastructure with the superstructure - and is placed on the boundary between them (see fig. 1). The purpose of this concept is to protect both structures and assure that will be no misunderstanding when people refer to equipments that are placed on the ***interstructure*** *level. The term refers especially to all communications and new technologies that merge with transportation area in the first place. Techniques like monitoring networks and implementing media (fiber optic, UTP [1], router [2] , switch [3] etc.) can be part of this concept.*

This definition relies on three fundamental concepts:

Infrastructure
n.
1. An underlying base or foundation especially for an organization or system.
2. The basic facilities, services, and installations needed for the functioning of a community or society, such as transportation and communications systems, water and power lines, and public institutions including schools, post offices, and prisons.

Infrastructural adj.
The term infrastructure has been used since 1927 to refer collectively to the roads, bridges, rail lines, and similar public works that are required for an industrial economy, or a portion of it, to function. The term also has had specific application to the permanent military installations necessary for the defense of a country. Perhaps because of the word's technical sound, people now use infrastructure to refer to any substructure or underlying system. Big corporations are said to have their own financial infrastructure of smaller businesses, for example, and political organizations to have their infrastructure of groups, committees, and admirers. The latter sense may have originated during the Vietnam War in the use of the word by military intelligence officers, whose task it was to delineate the structure of the enemy's shadowy organizations. Today we may hear that conservatism has an infrastructure of think tanks and research foundations or those terrorist organizations have an infrastructure of people sympathetic to their cause. [4]

Superstructure *n.*
1. A physical or conceptual structure extended or developed from a basic form.
2. The part of a building or other structure above the foundation.
3. The parts of a ship's structure above the main deck.
4. The rails, sleepers, and other parts of a railway. [5] [2]

Substructure *n.*
1. The supporting part of a structure; the foundation.
2. The earth bank or bed supporting railroad tracks. [6] [3]

Analyzing the above definitions we can frame ***interstructure*** between those known concepts as belong to each one into equal parts and in the same time to have their own roles.

3 Arguments for interstructure

A transportation telematic system could not exist without data and/or voice communication. The development of communications especially in transportation area depends on good architecture and implementation of research projects. [7] [4] If the communication architecture will be plan using the ***interstructure*** concept, then the standardized layers who give access to the information will be more accurately delimited.

[1] UTP - Unshielded Twisted Pair
[2] Router - A device that forwards data packets along networks. A router is connected to at least two networks, commonly two LANs or WANs or a LAN and its ISP's network. Routers are located at gateways, the places where two or more networks connect. http://www.webopedia.com/TERM/r/router.html
[3] http://www.webopedia.com/TERM/s/switch.html
[4] http://www.thefreedictionary.com/infrastructure
[5] http://www.thefreedictionary.com/superstructure , http://www.answers.com/topic/superstructure
[6] http://www.thefreedictionary.com/substructure
[7] M. Minea, F.D. Grafu, Transportations telematics - applications and fundamentals, Printech Publishing, Bucharest, 2005.

Let's go back now to the definition of ***interstructure***: when we say *"all components that assure the concomitant, directly or indirectly maintenance for a construction and for any entity that use it"*, we can consider an example from transportation area. If we think a video camera for surveillance of traffic as equipment dedicated to observe a specific area from a road, we can use this equipment for observe the carriageable, if there are splits in the road and we can interfere for redress the infrastructure or the superstructure. So, "video camera for traffic surveillance" is a part of ***interstructure***. In this moment all ITS [8] administrators are aware of difference between the superstructure of a specific location and the video camera that is not a part of that superstructure and is part of his level of action - ***interstructure***.

According to the most recently studies [9], important classification have been realized regarding to intelligent transport systems. In this studies appear intelligent infrastructure and intelligent vehicles as new terms. We include the following systems: traffic control systems, parking management systems, variable message signs, highway advisory radio etc., in ***interstructure***, so any equipment that offers better services for transportation using telecommunications can be included in our concept.

"The Internet is changing every aspect of our lives - education, research and more. Behind all this success is the underlying fabric of the Internet: the Internet Protocol (IP). IP was designed to provide best-effort service for delivery of data packets and to run across virtually any network transmission media and system platform. The increasing popularity of IP has shifted the paradigm from "IP over everything," to "everything over IP." In order to manage the multitude of applications such as streaming video, voice over IP, e-commerce, and others, a network requires quality of service (QoS) [10] in addition to best-effort service. Different applications have varying needs for delay, delay variation (jitter [11]), bandwidth, packet loss, and availability. These parameters form the basis of QoS. The IP network should be designed to provide the requisite QoS to applications. On the other hand, a file transfer application, based on ftp, doesn't suffer from jitter, while packet loss will be highly detrimental to the throughput [12] [13]" . [5] These new and advanced technologies will become normally technologies and will be used by anyone for day to day information. This part of communication technologies can be included too in the ***interstructure*** for tomorrow.

If we refer to the second definition of ***interstructure***: *"all equipments that assure a good functionality for infrastructure and superstructure through interworking of maintenance elements"*, we can think at ITS device servers that are facilitators of telematics, the new wave of technology for the transportation industry, this concept is a ***interstructure*** concept. Telematics is defined as data communications between systems and devices. It incorporates networked products in a vehicle, so a person can download information onto the central computer system, on board systems and sensors for driver protections etc. For example, a tire company can analyze tire performance for pressure, safety and environmental data. Data is collected on a pressure sensor mounted to the tire and submitted to the telematics module. All equipment that are not only a part of the vehicles but can connect with infrastructure and superstructure thought wireless or radio communication, or even Ethernet can be interstructure [14] [6]

On the basis of all applications transition toward IP we can accept that all telecommunication equipments must use this technology to be improved functional parameters. For example, a good decision in the future will be that all RS 232 interfaces descend to Ethernet medium. This is possible by adding networking equipment that enable remote access of semaphore, video camera and other monitoring equipment. A conclusive example of how information could be collected from traffic is represented below. [7]

4 Examples of what we can call interstructure

5.9 GHz DSRC (Dedicated Short Range Communications) is a short to medium range communications service that supports both public safety and private operations in roadside to vehicle and vehicle communication environments. DSRC is meant to be a complement to cellular communications by providing very high data transfer rates in circumstances where minimizing latency in the communication link and isolating relatively small communication zones are important. This DSRC could be interstructure. In ITS networks, users in vehicles expect that they are able to receive useful information continuously at any time, anywhere. Assumed ITS network model consist

[8] ITS - Intelligent Transportation Systems

[9] Information is from www.benefits.its.dov.gov

[10] QoS - Quality of Service

[11] Jitter is the deviation in or displacement of some aspect of the pulses in a high-frequency digital signal. As the name suggests, jitter can be thought of as shaky pulses. The deviation can be in terms of amplitude, phase timing, or the width of the signal pulse.

[12] http://www.sei.cmu.edu/str/indexes/glossary/throughput.html

[13] http://www.cisco.com/en/

[14] http://www.lantronix.com/solutions/transportation.html

of a backbone network and access networks. In this model, we consider two types of wireless communication infrastructure in access networks that are DSRC networks and cellular networks. Each access network has some local regions. A local region shows a hierarchical network, and is composed of multicasting routers, base stations that are located at the edge of the fixed network, and mobile hosts. Based stations are grouped into a subnet based on their proximity in the network. A designated server called a gateway server is positioned at the top of each local region. Mobile hosts move from one location to another across local regions and communicate with application servers through base stations while it is in its cell. Below it is shown the architecture [15]: [8]

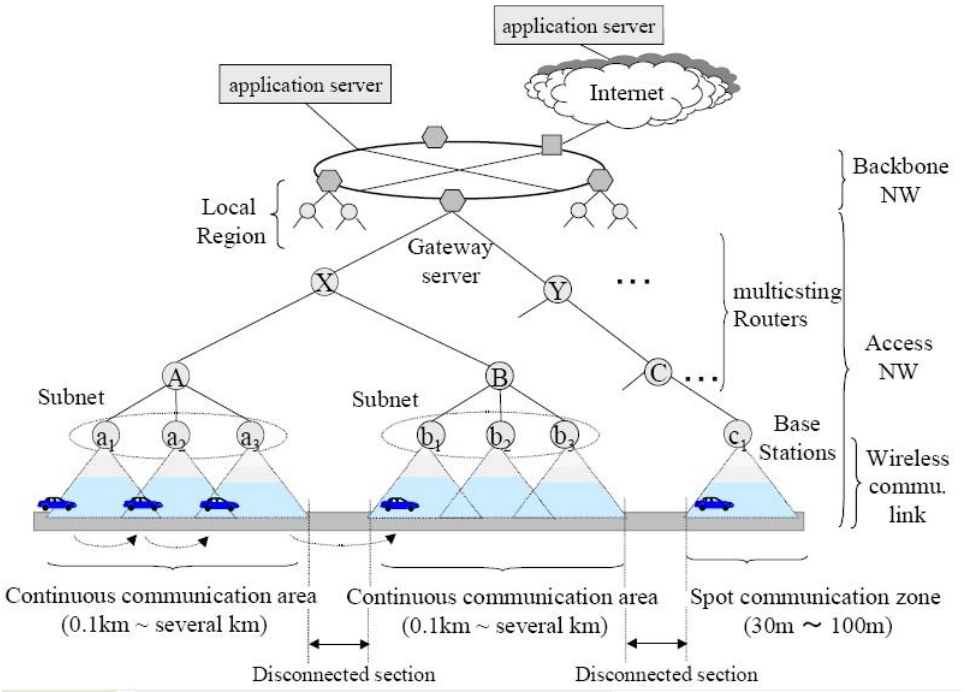


Figure 2: DSRC - based ITS network model

Building Ethernet connectivity into a product is no simple task. It requires a significant investment in hardware and software integration - often in areas outside of core competencies. Device networking starts with a device server. These solutions enable most any device with serial capability (TTL [16]) to become a fully functional member of an Ethernet network. Device servers include all of the elements needed to network-enable equipments - a processor, real-time operating system, a robust TCP/IP stack, a web server, and a network connection. All the connected product additionally needs is a header, providing connections to a power source and to a TTL serial port. The IP attributing to every serial device in transportation field to collect and offer information to traffic monitoring department and also to all agencies and persons involved in traffic will be an inherent advantage. Serial interfacing could be accomplished via a TTL connector, and for Ethernet access, an RJ 45 (10/100 Base -T) must be available. [17] [9] Rehearse this possible services sustain by IP and the most widely used medium as Ethernet we notice there are many details and aspects who can be included in interstructure concept: equipments that can accept a serial port and dynamic participates at the telecommunications process, networking devices (router, switch, repeater, fiber optic, etc) wireless devices that could not be considered infrastructure or superstructure etc. If we think and look at this communications field as an interstructure we can develop this area of transportation telecommunications more easy and for better understanding of this new domain we must understand NTCIP [18].

[15]Tatsushi Yamamoto, Advanced-grouped join mechanism for multicast group management in DSRC-based ITS networks, Faculty of Information, Shizuoka University, 2003

[16]TTL - Transistor-Transistor Logic

[17]http://www.lantronix.com/device-networking/embedded-device-servers/micro100.html

[18]NTCIP - The National Transportation Communications for ITS Protocol

5 Summary and Conclusions

The novelty is that the concept are analyzed through delimit a middle area between infrastructure and superstructure and call it interstructure, this concept line up behind ITS administrators and in the future could help telecommunication equipment administrators which, for example, design a communication support of a institution using it infrastructure and superstructure for obtain their interstructure who they can administrate as an independent entity against the first two. To make easy monitoring and manage the device servers, including all the elements needed, we must implement network-enable equipments and add IP to each device that has a serial port. Demand for bandwidth makes Ethernet the preferred protocol. This multitude of devices, new technologies and standards for transportation telecommunications we try to put in a single word interstructure.

References

[1] http://www.webopedia.com/

[2] http://www.thefreedictionary.com/

[3] http://www.answers.com/

[4] M. Minea, F. D. Grafu, “Transportations telematics - applications and fundamentals” Printech Publishing, Bucharest, 2005

[5] http://www.sei.cmu.edu/str/indexes/glossary/throughput.html

[6] http://www.cisco.com/en/us/tech/tk543/tk766/

[7] http://www.rad.com/article/

[8] Tatsushi Yamamoto, “Advanced-grouped join mechanism for multicast group management in DSRC-based ITS networks”, Faculty of Information, Shizuoka University, 2003

[9] http://www.lantronix.com/solutions/transportation.html

Florin Domnel Grafu
University POLITEHNICA of Bucharest, Romania
Department of Telematics and Electronics in Transports (Faculty of Transport)
Address: Splaiul Independentei, No. 313, Room JE 008, Bucharest
E-mail: grafffu@gmail.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 233-238

DDFS's Mathematical Approach Designing Considerations

Alin Grama, Lăcrimioara Grama

Abstract: This paper presents different methods for designing Direct Digital Frequency Synthesis (DDFS) circuits to recognize their advantages and disadvantages, and to identify which type is the best for different applications. DDFS is a very popular technique for generating stable frequencies with extremely precise resolution and fast switching speeds.

Keywords: DDFS circuits, Cordic, Phase to Amplitude Converter

1 Introduction

The paper is organized as follows. In Section 2 the classical architecture of DDFS is described. Next sections discuss some different types optimized for specific applications. In Section 3 an improved architecture is described based on a Phase Accumulator register and Phase to Amplitude Converter Lookup Table technique. Section 4 describes a DDFS designed using the Coordinate Rotation (CORDIC) algorithm.

2 Classic DDFS

The simplest form of a DDFS system can be implemented using a precision reference clock source, an address counter, a Read Only Memory (ROM) and a Digital to Analog Converter (DAC) [4]. This type of architecture is shown in Figure 1.

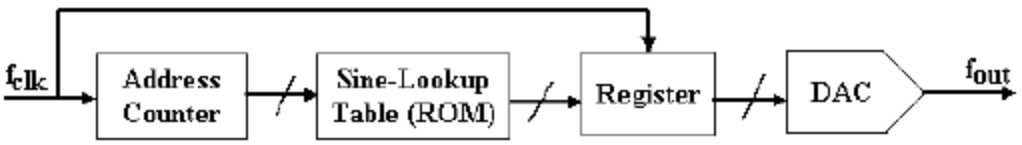


Figure 1: Classic DDFS architecture.

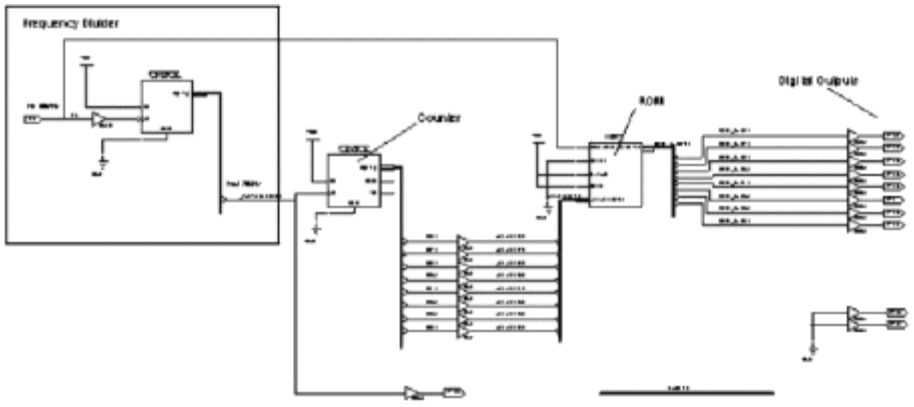


Figure 2: Electrical schematic of a classic DDFS.

The ROM contains the amplitude values of a complete period of the sine wave. The address counter strobe the memory (used like a lookup table) and the samples of the sine are present to the input of the Register. After that the samples are converted in analogic signal by a highest DAC converter. With this type of architecture, a good shape of sine can be obtained, but the disadvantage consists in impossibility to change the output frequency if the clock source is fixed. To test the capabilities of this kind of DDFS, we use an evaluation board equipped with FPGA chip Spartan II. The electrical schematic is shown in Figure 2.

In the final report we can see utilization of the device.

Device utilization summary:		
– – – – – – – – – –		
Selected Device: 2s50pq208 – 6		
Number of Slices:	19 out of 768	2%
Number of Slice Flip Flops:	17 out of 1536	1%
Number of 4 input LUTs:	8 out of 1536	0%
Number of bonded IOBs:	12 out of 144	8%
Number of BRAMs:	1 out of 8	12%
Number of GCLKs:	1 out of 4	25%

Table 1:

3 DDFS with phase accumulator

To obtain a more flexible device, we must to introduce a phase accumulator and an adder [3]. The counter is replaced with this phase accumulator. The output of the phase accumulator cannot be used to generate a waveform. After this accumulator we must place a phase to amplitude converter (lookup table). This type of architecture is shown in Figure 3, and the electrical schematic implemented on Spartan II FPGA device in Figure 4.

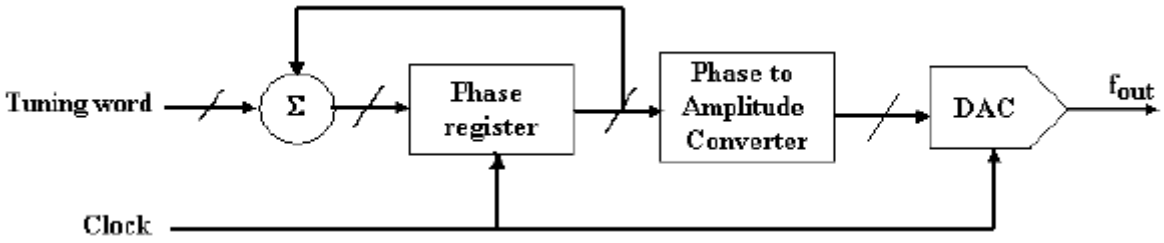


Figure 3: DDFS with phase register.

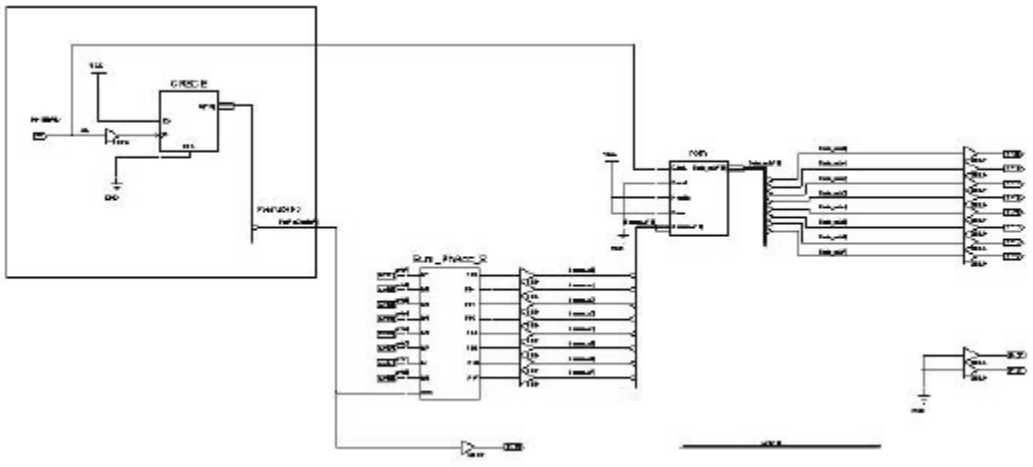


Figure 4: Electrical schematic of a DDFS with phase register.

Device utilization for this circuit are presented below:

This kind of circuit uses a little bit more of the resources of device, but the advantage is that we can obtain more values for the output frequencies. The frequency of output signal can be evaluated using the next equation:

$$f_0 = \frac{M f_c}{2^N}, \tag{1}$$

Device utilization summary:			
– – – – – – – – – – –			
Selected Device: $2s50pq208-6$			
Number of Slices:	20 out of	768	2%
Number of Slice Flip Flops:	17 out of	1536	1%
Number of 4 input LUTs:	8 out of	1536	0%
Number of bonded IOBs:	20 out of	144	13%
Number of BRAMs:	1 out of	8	12%
Number of GCLKs:	2 out of	4	50%

Table 2:

where M represents the tuning word, f_c the clock frequency, and N the dimension of phase register.

4 Cordic algorithm implementation

Another digital method used to synthesize frequencies is the Cordic (Coordinate Rotation) algorithm. This method is used for generating sinusoidal waveforms only, and it is based on trigonometric properties and transformations of sinusoidal and cosinusoidal functions [1].

We have to build up a sine of M Given's phase rotation stages of the form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \cos(\theta) \begin{bmatrix} 1 & -\tan(\theta) \\ \tan(\theta) & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2}$$

Each stage rotates the input complex number by $\pm\frac{\delta}{2^k}$ radians, where k is the k-th bit of the tuning word W, and $\delta = \frac{\pi}{2}$. The initial vector can be rotate in increments of $\frac{\delta}{2^M}$ radians, in the range $\left[0, \pi - \frac{\delta}{2^{M+1}}\right]$.

An angle θ can be assigned in this form:

$$\theta = \pi \sum_{k=2}^{M} a_k \frac{1}{2^i}, \tag{3}$$

where a_k is given by:

$$a_k = \begin{cases} 1 & \text{when the } k\text{-th bit of } w \text{ is equal to } 1, \\ -1 & \text{when the } k\text{-th bit of } w \text{ is equal to } 0. \end{cases} \tag{4}$$

When we set the angle θ at the value w, the rotation obtained is:

$$(x_k, y_k) = \begin{cases} x_{k-1} - T_k y_{k-1}, y_{k-1} + T_k x_{k-1} & \text{when} \quad W[k] = 1, \\ x_{k-1} + T_k y_{k-1}, y_{k-1} - T_k x_{k-1} & \text{when} \quad W[k] = 0, \end{cases} \tag{5}$$

where $T_k = \tan\left(\frac{\delta}{2^k}\right)$. The sinusoid can be obtained by continuously incrementing the tuning word W. The complete scheme is shown in Figure 5.

We identify the following advantages:

- only the sinusoidal waveforms can be synthesized;
- the number of stages determine the frequency resolution, so for a good resolution, more stages are needed;
- energy consumption is significant, because all of the sub-iteration stages have to operate at the clock frequency.

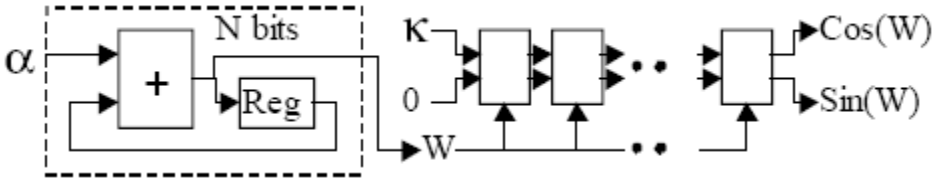


Figure 5: Cordic architecture.

5 ROM compression techniques

The dimension of the ROM can be reduced by use of sine/cosine symmetry. As you can see in Figure 6, the sine wave in the range $\left[0, \frac{\pi}{2}\right]$ is identical with the cosine wave in the range $\left[\frac{3\pi}{2}, 2\pi\right]$. Because of this property is not necessarily to store all the samples during the hall period of wave. It is enough to store only the samples from the range $\left[0, \frac{\pi}{2}\right]$, and to read more times the ROM content, in order to obtain the complete period of the sine wave. Only one-fourth of sine and cosine functions (from 0 to $\frac{\pi}{2}$) are stored in the memory. This method is better described in Figure 7.

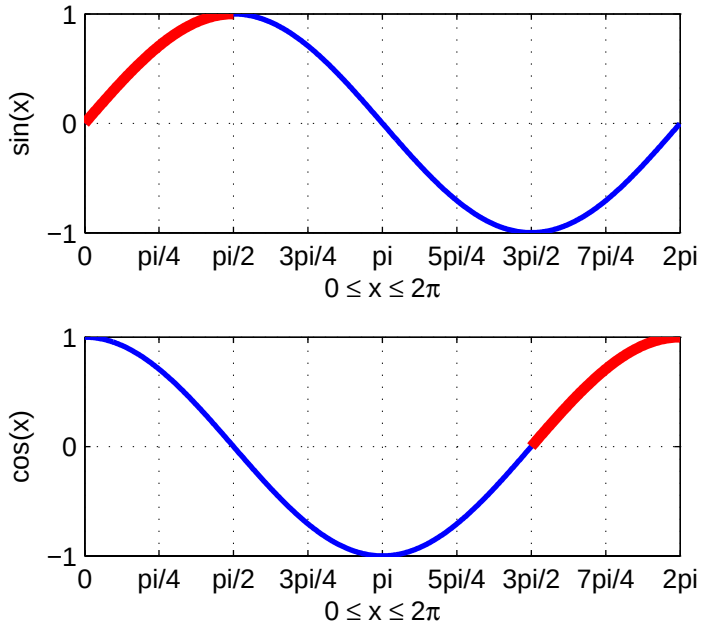


Figure 6: ROM compression principle.

In Table 3 is presented the content of a ROM that does not use the compression technique. This method represents an application of property of sine function, but it is not a signal processing application. A method based on signal processing application is the linear interpolation scheme, with the Taylor-series that implement the following equation:

$$\sin(\theta) = \sin(\theta_i) + \alpha(\theta + \theta_i) + \text{error}. \tag{6}$$

This simple technique was first introduced by Hutchison [5]. For this kind of implementation, we need two ROM with a total dimension smaller than the original ROM. In one memory (named "coarse" ROM) the values of $\sin(\theta_i)$ are stored, and in another ROM the interpolation coefficients (named "fine" ROM) are stored. In this manner, the size of the total ROM is reduced (a 50% reduction of the ROM size is possible), but the disadvantage consists in increasing the complexity of the system, by adding an adder and a multiplier. Sunderland has improved

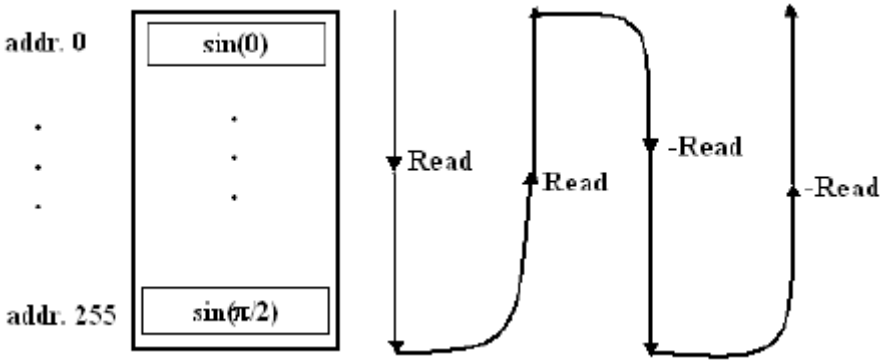


Figure 7: ROM compression implementation.

82*h*	84*h*	86*h*	88*h*	89*h*	*Bh*	8*Dh*	8*Fh*	91*h*	93*h*	95*h*	96*h*	98*h*	9*Ah*	9*Ch*	9*Dh*
9*Fh*	*A*1*h*	*A*3*h*	*A*4*h*	*A*6*h*	*A*8*h*	*A*9*h*	*ABh*	*ACh*	*AEh*	*AFh*	*B*1*h*	*B*2*h*	*B*4*h*	*B*5*h*	*B*6*h*
*B*8*h*	*B*9*h*	*BAh*	*BBh*	*BDh*	*BEh*	*BFh*	*C*0*h*	*C*1*h*	*C*2*h*	*C*3*h*	*C*4*h*	*C*5*h*	*C*6*h*	*C*6*h*	*C*7*h*
*C*8*h*	*C*8*h*	*C*9*h*	*CAh*	*CAh*	*CBh*	*CBh*	*CBh*	*CCh*	*CCh*	*CCh*	*CCh*	*CDh*	*CDh*	*CDh*	*CDh*
CDh	*CDh*	*CDh*	*CDh*	*CCh*	*CCh*	*CCh*	*CCh*	*CBh*	*CBh*	*CBh*	*CAh*	*CAh*	*C*9*h*	*C*8*h*	*C*8*h*
*C*7*h*	*C*6*h*	*C*6*h*	*C*5*h*	*C*4*h*	*C*3*h*	*C*2*h*	*C*1*h*	*C*0*h*	*BFh*	*BEh*	*BDh*	*BBh*	*BAh*	*B*9*h*	*B*8*h*
*B*6*h*	*B*5*h*	*B*4*h*	*B*2*h*	*B*1*h*	*AFh*	*AEh*	*ACh*	*ABh*	*A*9*h*	*A*8*h*	*A*6*h*	*A*4*h*	*A*3*h*	*A*1*h*	9*Fh*
9*Dh*	9*Ch*	9*Ah*	98*h*	96*h*	95*h*	93*h*	91*h*	8*Fh*	8*Dh*	8*Bh*	89*h*	88*h*	86*h*	84*h*	82*h*
7*Eh*	7*Ch*	7*Ah*	78*h*	76*h*	74*h*	72*h*	70*h*	6*Fh*	6*Dh*	6*Bh*	69*h*	67*h*	66*h*	64*h*	62*h*
60*h*	5*Fh*	5*Dh*	5*Bh*	5*Ah*	58*h*	56*h*	55*h*	53*h*	52*h*	50*h*	4*Fh*	4*Dh*	4*Ch*	4*Ah*	49*h*
48*h*	46*h*	45*h*	44*h*	43*h*	42*h*	41*h*	40*h*	3*Fh*	3*Eh*	3*Dh*	3*Ch*	3*Bh*	3*Ah*	39*h*	38*h*
38*h*	37*h*	36*h*	36*h*	35*h*	35*h*	35*h*	34*h*	34*h*	33*h*	33*h*	33*h*	33*h*	33*h*	33*h*	33*h*
33*h*	33*h*	33*h*	33*h*	33*h*	33*h*	33*h*	34*h*	34*h*	35*h*	35*h*	35*h*	36*h*	36*h*	37*h*	38*h*
38*h*	39*h*	3*Ah*	3*Bh*	3*Ch*	3*Dh*	3*Eh*	3*Fh*	40*h*	41*h*	42*h*	43*h*	44*h*	45*h*	46*h*	48*h*
49*h*	4*Ah*	4*Ch*	4*Dh*	4*Fh*	50*h*	52*h*	53*h*	55*h*	56*h*	58*h*	5*Ah*	5*Bh*	5*Dh*	5*Fh*	60*h*
62*h*	64*h*	66*h*	67*h*	69*h*	6*Bh*	6*Dh*	6*Fh*	70*h*	72*h*	74*h*	76*h*	78*h*	7*Ah*	7*Ch*	7*Eh*

Table 3: ROM content.

this technique, by splitting the phase of the sine function in three parts. The phase will be represented as:

$$\sin(A+B+C) = \sin(A+B)\cos(C) + \cos(A)\cos(B)\sin(C) - \sin(A)\sin(B)\sin(C), \tag{7}$$

where A represents the most significant bits, B the middle bits and C the LSBs (least significant bits). Because of the small size of B and C relatively to A, the ROM size is significantly reduced [2].

$$\sin(A+B+C) \approx sin(A+B) + \cos(A)\sin(C). \tag{8}$$

The resulting error in this method can be limited to 1 LSB, and the size of the ROM is reduced with the a factor by approximate 11. The method is implemented in the same architecture like in Hutchison's method, with a coarse ROM (for the first term) and a fine ROM table (for the second term). Of course, an adder is needed to sum the outputs of the coarse and fine ROM tables. An efficient method is an improvement of Sunderland's technique, known as sine-phase difference. In this method, the mean square error is reduced by using an optimization algorithm. Amplitude compression technique is used:

$$\left[\sin\left(\frac{\pi\phi}{2}\right) - \phi\right]. \tag{9}$$

Reducing the amplitude by 20% of its original value, two bits are saved for each word in the ROM lookup table.

6 Conclusions

Different methods to synthesize sinusoidal waveforms were presented with their advantages and disadvantages. This paper can be a start point for those engineers who want to design a DDFS because here are presented more possibilities of implementations.

References

[1] E. Grayver and B. Daneshrad, "Direct Digital Frequency Synthesis Using a Modified Cordic", *IEEE Trans. Comm.*, Vol. 47, No. 7 pp. 2255-2262, 1999.

[2] A. Bellaouar, M. O'brecht, A. Fahim and M. Elmasry, "Low-Power Direct Digital Frequency Synthesis for Wireless Communications", *IEEE Journal of Solid-State Circuits*, Vol. 35, No. 3, pp. 385-390, March 2000.

[3] A. Eltawil and B. Daneshrad "Interpolation Based Direct Digital Frequency Synthesis for Wireless Communications", *IEEE Trans. Comm.*, Vol. 50, pp. 73-77, 2002.

[4] W. Egan, *Frequency Synthesis by Phase Lock* , New York, John Wiley and Sons, Inc., 2000.

[5] B. Hutchison, *Frequency Synthesis and Applications*, New York, IEEE Press, 1975.

Alin Grama, Lăcrimioara Grama
Technical University of Cluj-Napoca
Applied Electronics Department
Address: Cluj, 24-26 Bariţiu Street, Cluj-Napoca, România
E-mail: Alin.Grama@ael.utcluj.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 239-244

Kramers-Kronig Relationship Computation by Gaussian Quadrature

Lăcrimioara-Romana Grama, Anca-Ioana Dişcant

Abstract: The Kramers-Kronig transformation has been extensively applied in optical spectroscopy to calculate the real component of an optical quantity from the imaginary component, such as the determination of the dispersive mode from a measurement of the absorptive mode and vice versa. In this paper, the Kramers-Kronig transformation is approximated.

Keywords: Kramers-Kronig transformation, specialized Gaussian quadrature, Cauchy principal value integrals, Hilbert transform

1 Introduction

The Kramers-Kronig transformation was developed independently by Kramers [1] and Kronig [2, 3], around 1927 and is widely used in optical spectroscopy [4, 8]. Hilbert transforms arise in many applications, and they are often called by different names such as dispersion relations, Kramers-Kronig transforms, and Cauchy principal value integrals. Because of the important applications of Hilbert transforms, considerable effort has been devoted to the numerical evaluation of Cauchy principal value integrals and there is also an extensive body of work devoted to the application of Kramers-Kronig transforms to experimental data. Some strategies have been focused on avoiding the principal value integrals altogether. A primary area of application is the analysis of optical data. There are two principal issues involved in optical data analysis [13]:

a. Fitting measurements to some particular functional form, including a resolution of the extrapolation problem to regions outside which spectral measurements have been made.

b. Solving the Kramers-Kronig inversion either analytically or numerically.

This study is concerned with detailing an approach to the numerical evaluation of Kramers-Kronig transform, which is fairly simple in application but gives rather good results. The easiest way to evaluate the Kramers-Kronig transformation is to use the Gaussian quadrature [8], in which the restriction of equally spaced evaluation points is dropped. The immediate effect is that the number of variables that can be used to optimize the evaluation of the integral are doubled, and another important thing is that the weights and abscissa values can be determined so that the quadrature is exact [10].

The paper is organized as follows. In Section 2 we will present the Kramers-Kronig relationship and in Section 3 the theoretical background of the Gaussian quadrature approach will be discussed. The approximation of the Kramers-Kronig transform using logarithmic Gaussian quadrature is described in Section 4 and an example of this is also presented in Section 5; the advantages of this specialized Gaussian approach are also underlined in Section 6.

2 Review of the Kramers-Kronig Relationship

If a complex function f is analytic at all points interior to and on a simple closed contour C, then $\oint_C f(z)dz = 0$. Let

$$f(z) = g(z) + jh(z) = g(x,y) + jh(x,y), \tag{1}$$

where $z = x + jy$ ($g(x,y)$ and $h(x,y)$ are real valued functions), such that $f(z)$ is analytic and $f(z) \to a$ as $|z| \to \infty$ in either the lower or upper half of the complex plane (a is a constant which may be complex). We will assume here that this is true on the upper half of the complex plane. Consider the integral:

$$\oint_C \frac{f(z)-a}{z-\omega}dz, \quad C \text{ is the closed curve (Figure 1).} \tag{2}$$

Since $f(z)$ is analytic and $z - \omega \neq 0$ on or interior to C, $(f(z)-a)/(z-\omega)$ is also analytic on and interior to C, and by the Cauchy integral theorem the integral over this path is 0. Considering the limit as $\rho \to \infty$, the integral

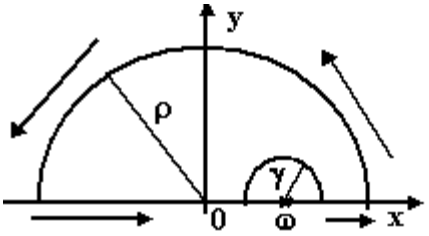


Figure 1: The closed curve, C, used to prove the Hilbert transform.

over this part of the curve is 0 ($f(z)-a \to 0$) [4]. The sum of the integral over the real axis and the semicircle centered about ω must be also 0:

$$\int_{-\infty}^{\omega-\gamma} \frac{f(x)-a}{x-\omega}dx + \int_{\omega+\gamma}^{\infty} \frac{f(x)-a}{x-\omega}dx + j\int_{x}^{0} \Big(f(\omega+\gamma e^{j\theta})-a\Big)d\theta = 0. \tag{3}$$

Now consider the limit of this equation as $\gamma \to 0$:

$$P.V.\int_{-\infty}^{\infty} \frac{f(x)-a}{x-\omega}dx - j\pi(f(\omega)-a) = 0, \quad P.V. \text{ - Cauchy principal value of the integral [7].} \tag{4}$$

Separating the real and imaginary parts, the result is:

$$g(\omega)-Re(a) = \frac{1}{\pi}P.V.\int_{-\infty}^{\infty} \frac{h(x)-Im(a)}{x-\omega}dx, \quad h(\omega)-Im(a) = -\frac{1}{\pi}P.V.\int_{-\infty}^{\infty} \frac{g(x)-Re(a)}{x-\omega}dx. \tag{5}$$

If we consider the special case when a is a real quantity (thus $Re(a) = g_\infty$ and $Im(a) = 0$), we will have:

$$g(\omega)-g_\infty = \frac{1}{\pi}P.V.\int_{-\infty}^{\infty} \frac{h(x)}{x-\omega}dx, \quad h(\omega) = -\frac{1}{\pi}P.V.\int_{-\infty}^{\infty} \frac{g(x)-g_\infty}{x-\omega}dx. \tag{6}$$

These equations are called the Hilbert transforms [6]. If $f(x) = f^*(-x)$ then Eqs. (5) become:

$$g(\omega)-Re(a) = \frac{2}{\pi}P.V.\int_{0}^{\infty} \frac{x(h(x)-Im(a))}{x^2-\omega^2}dx, \tag{7}$$

$$h(\omega)-Im(a) = -\frac{2\omega}{\pi}P.V.\int_{0}^{\infty} \frac{g(x)-Re(a)}{x^2-\omega^2}dx. \tag{8}$$

These are known as the Kramers-Kronig transforms in the electrical engineering. The Kramers-Kronig is a special case of Hilbert transform. It should also be noted that the real part need only be even and the imaginary part need only be odd on the real axis, not over the entire upper half of the complex plane. The requirements for the Kramers-Kronig transform to hold, can be summarized as follows:

a. The function must be analytic over the upper half (or lower half) of the complex plane (or the sum of the residues at the singular points must be 0), and the function must go to a real constant as the complex variable z goes to infinity.

b. The real part of the function must be even and the imaginary part must be odd on the real axis.

The Hilbert transform, given by Eqs. (6), only requires the first criterion.

3 Gaussian Quadrature Approach

The approach to be employed involves the use of specialized quadrature procedures. A general method for the numerical evaluation of integrals is a Gaussian quadrature [10, 11]. The classical formula for the numerical evaluation of an integral takes the form:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} w_i f(x_i), \tag{9}$$

where N is the number of sample points in the interval (it can be open or closed), x_i denotes the points at which the integrand is sampled, and the w_i represent weighing coefficients at the sampling points. The simplest examples of this approach are the trapezoidal and Simpson's rules (the abscissa values, x_i, must be selected in an equally spaced fashion).

In Gaussian quadrature schemes, the restriction of equally spaced evaluation points is dropped. This has the immediate effect of doubling the number of variables that can be used to optimize the evaluation of the integral. The second feature of considerable importance is that the weights and abscissa values can be determined so that the quadrature is exact (to approximately whatever machine precision is being employed) for integrands of the form:

$$f(x) = W(x)p(x), \tag{10}$$

where $p(x)$ is a polynomial and $W(x)$ denotes a weight function ($W(x) \geq 0$ for $x \in [a,b]$). If the weights and abscissa values are specifically tailored for the function $f(x)$, we can write:

$$\int_a^b f(x)dx = \int_a^b W(x)p(x)dx \approx \sum_{i=1}^{N} w_i p(x_i). \tag{11}$$

The approximation sign is maintained, since we are concerned with computer evaluations. One important observation is that the function $W(x)$ no longer occurs explicitly in the summation in Eq. (11), but appears implicitly in the values of $\{w_i, x_i\}$. Eq. (11) is exact when $p(x)$ is a polynomial up to order $2N-1$.

It is possible for a variety of common functional forms to tabulate $\{w_i, x_i\}$ for different values of N, assuming that the integration range is kept fixed. This has been done for the functions shown in Table 1 [12]. This list defines the standard weight functions employed in Gaussian quadrature (functions not in this list give rise to what are generally termed specialized Gaussian quadratures).

Integration range	Weight function $W(x)$	Name
$[-1,1]$	1	Gaussian quadrature (Gauss-Legendre quadrature)
$[-1,1]$	$(1-x)^{\alpha}(1+x)^{\beta}$	Gauss-Jacobi quadrature
$[-1,1]$	$\frac{1}{\sqrt{1-x^2}}$	Gauss-Chebyshev quadrature
$[0,\infty)$	$\exp(-x)$	Gauss-Laguerre quadrature
$(-\infty,\infty)$	$\exp(-x^2)$	Gauss-Hermite quadrature

Table 1: Some common functions and the integration ranges for which $\{w_i, x_i\}$ are available as a function of N.

4 Specialized Gaussian Quadrature: Kramers-Kronig Transform

For the analysis of functions that have a particular even or odd symmetry and are measured as a function of a variable that takes on positive values (i.e., a frequency), it is more common to write the Hilbert transforms as given in Eqs. (7) and (8). We can also write Eq. (8) as:

$$(Hf)(x) = \int_0^1 \log(s^{-1})K(s,x)ds \quad \text{for} \quad f(x) \quad \text{even}, \tag{12}$$

with $K(s,x)$ given by

$$K(s,0) = 0, \quad K(s,x) = \frac{1}{\pi}\{f'[x(1-s)] - f'[x(1+s)] + f'[x(s^{-1}-1)] - f'[x(s^{-1}+1)]\} \quad \text{for} \quad x \neq 0. \tag{13}$$

The Hilbert transforms given in Eqs. (7) and (8) can each be reduced to two mathematically equivalent Kramers-Kronig transform relations. The choice of one form over the other is dictated by the structure of the function near the origin. For f even a result closely related to Eq. (12) can be readily derived in which the kernel function has a slightly different form from that given in Eq. (13). For Eq. (7) we obtain:

$$(Hf)(x) = \int_0^1 \log(s^{-1})K_1(s,x)ds \quad \text{for} \quad f(x) \quad \text{odd}, \tag{14}$$

where $K_1(s,x)$ is given by

$$K_1(s,x) = \frac{1}{\pi x}\{g'[x(1-s)] - g'[x(1+s)] + g'[x(s^{-1}-1)] - g'[x(s^{-1}+1)]\} \quad \text{for} \quad x \neq 0, \tag{15}$$

where $g(s) = sf(s)$, and next equation applies for $x = 0$:

$$K(s,0) = -2\frac{1}{\pi}[f'(s) + f'(s)^{-1}]. \tag{16}$$

The weight function described in Section 3 is identified with $\log(s^{-1})$. The result for the Hilbert transform is:

$$(Hf)(x) = \sum_{i=1}^{N} w_i K(x_i,x), \tag{17}$$

where the weights (w_i) and evaluation points (x_i) are determined from the set of polynomials based on the weight function $\log(s^{-1})$. The principal advantage of this approach is that the singularity in the original integral is now incorporated in the weights w_i. The one possible numerical problem that might occur is a loss of precision in the evaluation of $K(x_i,x)$. Some numerical experiments with representative functional forms indicates that this problem does not arise to any significant extent [13].

The principal applications of the use of Eq. (17) fall into two main groups. The first are those problems for which the function is specified, but the Hilbert transform cannot be evaluated in terms of known functions. The second group of examples comprises those cases for which the function is unknown but is instead represented by a set of discrete experimental data points [9].

5 Example

We will present an example that can be evaluated in a simple closed form. Suppose a set of data, which has the form of a set of discrete points $\{A_i, x_i\}$, fitted to a Lorentzian line profile. The Lorentzian function is

$$A(x) = \frac{1}{\pi}\frac{a}{a^2 + (x-x_0)^2}, \tag{18}$$

where a and x_0 are constants. The factor of π^{-1} in Eq. (18) is selected so that the Lorentzian encloses unit area on the interval $(-\infty,\infty)$. The Lorentzian can also be normalized so that the curve encloses unit area on the interval $[0,\infty)$. On this interval we might consider A_i as an absorption intensity and x_i as a frequency. The Kramers-Kronig transform of the Lorentzian can be obtained in closed form:

$$(HA)(x) = \frac{1}{\pi}\frac{(x-x_0)}{a^2 + (x-x_0)^2}, \tag{19}$$

Lets evaluate the Gaussian quadrature approach for the Lorentzian, using Eq. (17). The result is:

$$(HA)(x) = \sum_{i=1}^{N} w_i K(x_i,x). \tag{20}$$

In Table 2 we show a comparison of the use of the quadrature formula versus the exact result as a function of x. The calculations were carried out in a precision by use of 20 digits[1]. If we are dealing with experimental data, typically no more than three to four digits of precision for the data are typically available, and therefore an error of $\sim 10^{-2}$% in the Kramers-Kronig transformation would be acceptable. Except at $x = 1$, this condition is met. The error at $x = 1$ is governed by machine roundoff. The exact value for the Kramers-Kronig transform at this frequency is zero, and the calculated quadrature value is -3×10^{-30}, which is in excellent agreement with the true result.

In Figure 2 is presented the original function, its Kramers-Kronig transformation and also the approximation of its Kramers-Kronig transformation. The last subplot represents the percentage error between the Kramers-Kronig transformation and its approximation, which provide that the approximation has very good results.

[1]The percentage error is given as the difference between the logarithmic Gaussian quadrature approximation in $N = 30$ points and the exact result of the Kramers-Kronig transformation divided by the approximated value of the Kramers-Kronig relationship, using 20 iterations.

x	Kramers-Kronig transform	Percentage error $N = 10$	Percentage error $N = 30$
0.1	$-.15827563401403957343$	$.24245264594614771014$	$.24036632852811657171e-4$
0.5	$-.12732395447351627093$	$.13086561198888666576e-5$	$.43598356225108015373e-15$
0.9	$-.31515830315226790736e-1$	$-.33420522697829840565e-10$	$-.36328330324571233913e-15$
1.0	.0	$---$	$---$
5.0	$.74896443807950754956e-1$	$-.11197209341695572946e-2$	$.25528819114439382458e-10$
10.0	$.34936450922611168856e-1$	$-.22513384878386000220e-1$	$.30080475354495508758e-6$
50.0	$.64934156631997275513e-2$	$.26386221546937138615$	$.50655938294708368133e-2$

Table 2: Kramers-Kronig transform of the Lorentzian by use of a logarithmic Gaussian quadrature (the values $a = 1$ and $x_0 = 1$ have been employed).

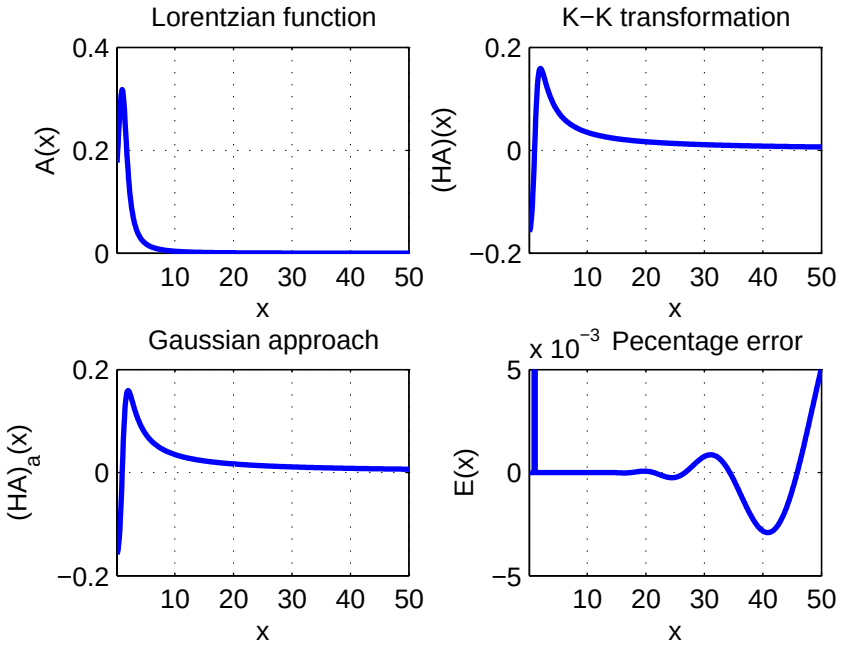


Figure 2: The Lorentzian function, its Kramers-Kronig transformation, the approximated Kramers-Kronig transformation (using logarithmic Gaussian quadrature - $N = 30$), and the corresponding percentage error.

6 Summary and Conclusions

From the results above it appears that values of around $N = 30$ are sufficient to obtain the Kramers-Kronig transformation to a precision that is better or approximately matches the experimental precision. Since the computer time required for the numerical evaluation of Eq. (8) is almost negligible, the safest approach is to employ the largest size quadrature possible, assuming the availability of the abscissa points and weights.

In summary, it has been found that the proposed procedure for numerical evaluation of Kramers-Kronig transforms yields results of high precision. The ease of implementation makes the proposed technique attractive as a means for numerical evaluation of these singular integrals. All the intensive computational labor occurs in the determination of the weights and evaluations points, but this needs to be done only once. The evaluation points were compute using Mathematica 5.2.

References

[1] H. A. Kramers, "La diffusion de la lumiére par les atomes", *Atti Congr. dei Fisici*, Como 2, pp. 545-557, 1927.

[2] R. de L. Kronig, "On the Theory of Dispersion of X-Rays", *J. Opt. Soc. Am. Rev. Sci. Instrum.*, Vol. 12, pp. 547-557, 1926.

[3] C. J. Gorter and R. de L. Kronig, "On the Theory of Absorption and Dispersion in Paramagnetic and Dielectric Media", *Physica III*, Vol. 9, pp. 1009-1020, 1936.

[4] K. Yamamoto and H. Ishida, "Optical Theory Applied to Infrared Spectroscopy", *Vib. Spectrosc.*, Vol. 8, pp. 1-36, 1994.

[5] Jun Ye, L. S. Ma and John L. Hall, "Using FM methods with Molecules in a High Finesse Cavity: A demonstrated path to $< 10^{-12}$ Absorption Sensitivity", in *Cavity-Ringdown Spectroscopy: An Ultratrace-Absorption Measurement Technique*, K. Busch and M. Busch, Eds., American Chemical Society/Oxford University Press, Washington, D.C., p. 233, 1999.

[6] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.

[7] K. Diethelm, "Peano Kernels and Bounds for the Error Constants of Gaussian and Related Quadrature Rules for Cauchy Principal Value Integrals", *Numer. Math.*, Vol. 73, pp. 53-63, November 21 1996.

[8] D. Y. Smith, "Dispersion Theory, Sum Rules, and their Application to the Analysis of Optical Data", in *Handbook of Optical Constants of Solids*, E. D. Palik, ed.,Academic, Orlando, Fla., pp. 3568, 1985.

[9] K. E. Peiponen, E. M. Vartiainen, and T. Asakura, *Dispersion, Complex Analysis and Optical Spectroscopy*, Springer- Verlag, Berlin, 1999.

[10] "Numerical Receipts in C: the Art of Scientific Computing", http://www.nr.com: sample pages, Copyright (C) 1988-1992 by Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Receipts Software.

[11] A. H. Stroud and D. Secrest, "Gaussian Quadrature Formulas", Prentice-Hall, Englewood Cliffs, N. J., 1966.

[12] B. P. Demidovich and I. A. Maron, *Computational Mathematics*, Moscow: English translation, Mir Publishers, 1981.

[13] F. W. King, "Effcient Numerical Approach to the Evaluation of Kramers-Kronig Transforms", *J. Optical Society of America B*, Vol. 19, pp. 2427-2436, October 2002.

Lăcrimioara-Romana Grama, Anca-Ioana Dişcant
Technical University of Cluj-Napoca
Basis of Electronics Department
Address: 24-26 Bariţiu Street, Cluj-Napoca, România
E-mail: Lacrimioara.Grama@bel.utcluj.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 245-250

Distributed Machine Learning in a Medical Domain

Horea Adrian Grebla, Calin Ovidiu Cenan

Abstract: Data-mining is concerned with extracting knowledge from databases (in this case we deal with distributed ones) using machine learning techniques. Traditionally, data-mining systems are designed to work on a single data set. However, with the increasing number of distributed database dispersed over many machines in WANS with geographically spread locations it is necessary to adopt new techniques to improve the overall system response. The development of Bayesian belief networks and associated algorithms made possible that probabilistic reasoning becomes a real option for a large variety of Artificial Intelligence applications. In this paper we present a methodology for machine learning using Bayesian belief network with practical exemplification for predicting arteriosclerosis cardiovascular disease.

Keywords: distributed data mining, Bayesian belief networks, machine learning, distributed databases

1 Introduction

Firms are concerned in increasing the flexibility of developing applications using standards to achieve interoperability, and to manage their infrastructure resources (processors, networks, storage, applications) efficiently by taking advantage of new business models and system management techniques, including distributed databases. Enterprises are adopting new approaches to distributed computing to meet earlier and nowadays needs. The traditional problems include legacy integration and adapting to changes in platforms and computing environments. New problems arise from business restructuring, outsourcing and new possibilities (services offered by new vendors). Enterprises must also handle effort demanding requirements, such as supporting peak loads for technical computing and running massive farms of web servers. We present a set of guiding principles that must apply to a system designed for modern WAN environments. The practical exemplification suits well the growing needs for the healthcare domains, where world gathered information can lead to better treatment, to a decreasing number of fatalities, and so to a healthier society. In the medical field the research results and the cures for critical diseases do not spread fast and wide enough compared to increasing number of patients so a world wide distributed database system can improve medical results.

2 Distributed Database Design Issues

Distributed database management system [12] relies on decisions for data placement over a computer network and also has to ensure local applications for each computational component as well as global applications on more computational machines; it has to provide a high-level query language with distributed query power, for distributed applications development.

To improve the performance of global queries, data can be partitioned and spread over the systemŠs components. Each network node executes applications as well as database management functions.

The requirements of wide-area distributed database systems differ dramatically from those of LAN systems. In a WAN configuration , individual sites usually report to different system administrators, have different access and charging algorithms, install site-specific data type extensions, and have different constraints on servicing remote requests. Typical of the last point are production transaction environments, which are fully engaged during normal business hours, and cannot take on additional load. Finally, there may be many sites participating in a WAN distributed DBMS.

A distributed database system supports data fragmentation if a relation stored within can be divided in pieces called fragments. These fragments can be stored on different sites residing on the same or different machines. The aim is to store the fragments closer to where they are more frequently used in order to achieve best performance. The partitions can be created horizontal, vertical or mixed [13] (the combination of horizontal and vertical fragmentation).

Let $\Re[A_1, A_2, ..., A_n]$be a relation where $A_i, i = \overline{1,n}$ are attributes. A horizontal fragment can be obtained by applying a restriction: $\Re_j = \sigma_{cond_j}(\Re)$, where $cond_j$ is the guard condition. So we can rebuild the original relation by union as follows:

$$\Re = \Re_1 \cup \Re_2 \cup ... \cup \Re_k.$$

A vertical fragment is obtained by a projection operation:

$$\Re_j = \prod_{\{A_{x1}, A_{x2}, ..., A_{xp}\}} (\Re),$$

where $Ax_i, i = \overline{1, p}$ are attributes. The initial relation can be reconstructed by join of the fragments:

$$\Re = \Re_1 \otimes \Re_2 \otimes ... \otimes \Re_l$$

The manner in which data are physically stored should not corrupt the way the queries and programs are written. This property, known as data independence, is now considered for heterogeneous distributed database, or multi-database. From the perspective of information exchange we consider the DBMS independent approach, which allows construction of open systems, with good scaling capabilities in heterogeneous environments.

3 Bayesian belief networks

The development of Bayesian belief networks and associated algorithms as presented by Pearl [01] has made proper probabilistic reasoning a real option for a large variety of Artificial Intelligence applications. Bayes nets are rapidly becoming a tool of choice for applied AI.

In general, probability calculations are computationally intractable [02], fact that led early expert systems to use a variety of simplifications, including certainty factors, fuzzy logic or other techniques. These systems, from the probabilistic point of view, all invoke strong independence constraints. Bayes nets can represent arbitrary probability distributions, and so overcome the limits of these independence assumptions and at the same time can take advantage of any simplifications possible given any real independence in the application domain. Bayesian belief networks provide a natural causal interpretation and so are fruitful methods for building expert systems with human input. They can easily be made to combine both expert knowledge and data mining.

There have been many medical applications of Bayesian belief networks [06], [07], and [08], however those projects focus on automatic expert diagnosis, not on discovering new connections and maybe none applying data mining methods to arteriosclerosis primary prevention. In this paper, we look at such an application to medical data set (STULONG) obtained from European Center for Medical Informatics, Statistics and Epidemiology, euromise.vse.cz. The data sets contain information about two sites participated in the several years of observations, namely Institute of Clinical and Experimental Medicine in Prague and Medical Faculty of the Charles University in Plzen.

The epidemiological data, specifically assessment of risk for arteriosclerosis cardiovascular disease given previous conditions. The aim of our study was to identify arteriosclerosis risk factors prevalence in a population generally considered to be the most endangered by possible arteriosclerosis complications, i.e. middle aged men. A possible extension of this work will be to consider the impact of complex risk factors intervention on their development and cardiovascular disease evolution. Superior predictions of such disease would allow for better allocation of health care resources and improved outcomes. Amid blossoming healthcare costs, cost effectiveness has become a dominating consideration for determining which preventive strategies are most appropriate.

A Bayesian belief network is a graph with arcs connecting nodes and no directed cycles (a so called directed acyclic graph), whose nodes represent random variables and whose arcs represent direct dependencies. Each node has a conditional probability table, which, for each combination of values of the parents, gives the conditional probability of each of its values. Users can set the values of any combination of nodes in the network that they have observed. This evidence propagates through the network, producing a new probability distribution over all the variables in the network. There are a number of efficient exact and approximate inference algorithms for performing this probabilistic updating [05], providing a powerful combination of predictive, diagnostic and explanatory reasoning.

Figure 1 shows a very simple example of Bayes nets for a medical diagnosis problem, namely for metastatic cancer, with the corresponding conditional probability table. Metastatic cancer is a possible cause of brain tumors and is also an explanation for increased total serum calcium. In turn, either of these could explain a patient falling into a coma. Severe headache is also associated with brain tumors.

Specifying the value of an observed variable would provide us with a revised diagnosis and expected value for all the other variables. Figure 2 shows the updated belief given the observation the patient is in a coma.

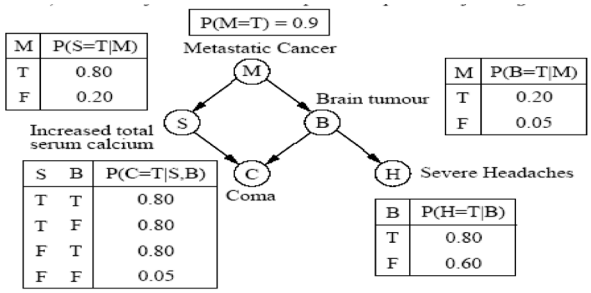


Figure 1: Example of Bayesian belief network

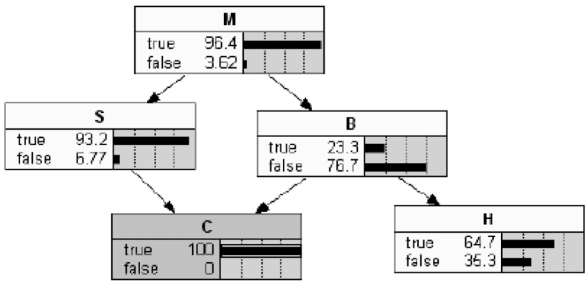


Figure 2: Belief revision

Bayesian belief networks can be either handcrafted or machine learned. Traditional knowledge engineering methods build expert systems by handcrafting the Bayes nets to match expert knowledge. This is a very slow and tedious process, so that it has been called the knowledge bottleneck. Machine learning methods are necessary to break through the bottleneck. Fully automated methods learn both the structure and the probability parameters.

4 Learning Bayesian belief networks

The goal of data mining is to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.

We have built Bayesian belief network for predicting arteriosclerosis cardiovascular disease by applying WEKA, a causal discovery program described in [03], and then we evaluated these Bayes nets using JavaBayes [04].

The dual nature of a Bayesian belief network makes learning in this case a natural division in two stage process: first learn a network structure, then learn the probability tables. Once a good network structure is identified, the conditional probability tables for each of the variables can be estimated.

WEKA attempts to learn the best causal structure to account for observational data, using an estimator metric and a search algorithm over the model space.

The dual nature of a Bayesian network makes learning of Bayes nets as a two stage process a natural division: first learn the network structure, then learn the probability tables. Once a good network structure is identified, the conditional probability tables for each of the variables can be easily estimated.

There are various estimator approaches to structure learning in Bayesian belief network learning. In WEKA, the following are presented: Local score metrics - Learning a network structure can be considered an optimization problem where a quality measure of a network structure given the training data needs to be maximized. The quality measure can be based on a Bayesian approach, minimum description length, information and other criteria. Those metrics have the practical property that the score of the whole network can be decomposed as the sum (or product) of the score of the individual nodes. This allows for local scoring and thus local search methods. Conditional independence tests - These methods stem from the goal of uncovering causal structure. The assumption is that there is a network structure that exactly represents the independencies in the distribution that generated the data. Then it follows that if a (conditional) independency can be identified in the data between two variables that there is no arrow between those two variables. Once locations of edges are identified, the direction of the edges is assigned such that conditional independencies in the data are properly represented. Global score metrics - A natural way to measure how well a Bayesian belief network performs on a given data set is to predict its future performance by estimating expected utilities, such as classification accuracy. Cross validation provides an out of sample evaluation method to facilitate this by repeatedly splitting the data in training and validation sets. A Bayesian belief network structure can be evaluated by estimating the network's parameters from the training set and the resulting Bayes nets performance determined against the validation set. The average performance of the Bayesian belief network over the validation sets provides a metric for the quality of the network. Cross validation differs from local scoring metrics in that the quality of a network structure often cannot be decomposed in the scores of the individual nodes. So, the whole network needs to be considered in order to determine the score. For each of these estimators, different search algorithms are implemented in WEKA, such as hill climbing, simulated annealing and tabu search.

The Bayesian information-theoretic metric, figure a tradeoff between model complexity and goodness of fit, thereby avoiding overfitting the data. Using our experiments we can conclude that this method has matched the best alternative machine learning algorithms across a range of problems. In order to obtain this result we used the

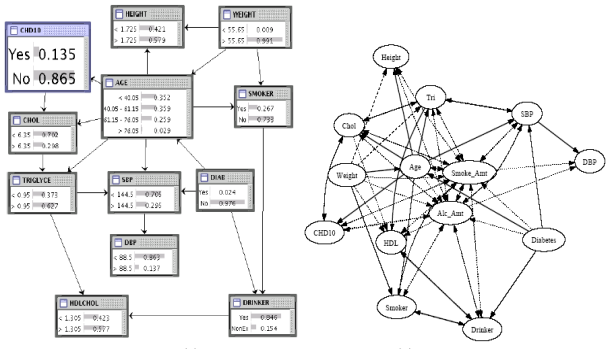


Figure 3: After data mining domain

part of WEKA allowing systematic experiments to compare Bayesian belief network performance with general purpose classifiers like C4.5, nearest neighbor, support vector, etc.

First we ran WEKA on the whole data set, using different estimators and searching methods and although there are some variations we note that arteriosclerosis is never directly connected to SBP, Diabetes, or Smoking, which are related only through other variables and conversely is always linked to Age, which has a strong effect, and also HDL and LDL.

Looking for smaller and more comprehensible results we use data for males only, omitting sex in order to have still a valid model of the reality. Figure 3 shows the results and we can see that in the summary model, given Age and Total Cholesterol, the new variables do not help very much for predicting arteriosclerosis (the CHD node). However, in the best model, drinking raises HDL (good cholesterol). Intervening on Drinking (to prevent additional correlation from the back paths through Age) we find the probability of high HDL is 0.4 for nondrinkers, but 0.6 for drinkers. However, in this model, that will have no effect on anything else, because there are no variables downstream of HDL.

So far in our experiments we have obtained some Bayesian belief networks, knowledge engineered by learning from data sets and we want to see if they do not overfit the data, leading to poor generalization? We partitioned the data into n sets of equal size, and in turn make each the test set for models learned on the remaining n - 1, giving us n experiments. In cases in which the value of n is too small we have little confidence that our runs are representative; if too large we introduce too much correlation among the training sets for each of the n runs, misleading us into thinking we have less variance than we really do. We conducted our experiments and obtained positive results with 10 fold cross-validation and to further reduce variance we use stratified samples, meaning that the 10 folds all have about the same proportion of positive and negative examples. This is roughly equivalent to stratified random samples in clinical trials.

A second question we want to answer during our experiments was how good are the inference methods which gave us those models? In other conducted experiments we compared our discovered model against standard machine-learning algorithms provided by WEKA: J48 (C4.5), logistic regression, and an artificial neural network all run with default settings.

We found that the our method does as well as a logistic regression model of the data set, which is otherwise the best model. Our discovered Bayesian belief network come second performing about the same and we can also state that J48 did quite poorly.

5 A proposed architecture for distributed machine learning

Knowledge discovery (or data-mining) is concerned with extracting knowledge from databases (possible distributed) using machine learning techniques. Traditionally, data-mining systems are designed to work on a single data set. However, with the growth of distributed database is increasingly dispersed over many machines in many different geographical locations.

Software agents [10] are one response to the problem of using the vast amounts of information stored on distributed sites. There are many types of software agents [11]; however, agents are typically thought of as being intelligent programs, which have some degree of autonomy. We intend to design an open, flexible data-mining agent architecture. A group of such agents will be able to cooperate to discover knowledge from distributed

sources.

A group of intelligent agents may work together in order to solve a problem or achieve a common goal. In order to do this they can use machine learning techniques in order to refine their knowledge. Learning in an intelligent system should improve the performance of that system. Our proposed research is concerned with how they can do this effectively.

Our high-level model is as follows. One or more agents per site are responsible for examining, analyzing and learning from local data sources. After this the agents will integrate the new knowledge they produced into a globally coherent theory using a supervisory agent, responsible for coordinating the discovery agents.

There are many ways learning can occur when data is distributed. The approach we proposed is for the agents to learn locally, and then to share their results, which are then refined and integrated by other agents in light of their own data and knowledge. This model permits the use of standard algorithms, and also allows inter-operation between different algorithms. We are adopting this approach, as it provides distributed processing together with flexibility in deploying off-the-shelf algorithms. The main problem here is how to integrate the local results.

There are different distributed learning methods but we favor a simple knowledge integration. Each agent learns a local theory and the resulting theories are tested against all the training examples, the best theory is selected and is then compared with the test set. In [09] is clearly demonstrated that for certain data sets simple knowledge integration is the best method for implementing distributed learning.

6 Summary and Conclusions

We have made an initial effort in Bayesian belief network modeling of medical data, developing a set of networks using a causal discovery algorithm. We found that the obtained Bayes nets does as well as a logistic regression model of the data set, which is otherwise the best model.

When run in Bayesian belief network modeling software (e.g., JavaBayes), these models provide a simple to use GUI and they tell an intuitive causal story of arteriosclerosis risk. JavaBayes allows also for the modeling of decision making, such as medical and health costs, and the modeling of health interventions. Thus, there is significant potential in providing the medical community with a useful set of tools for assessing arteriosclerosis risk and potential benefits from interventions.

There is also plenty of scope for future data mining efforts in the field of distributed data sets. We propose an agent-based architecture to distributed knowledge discovery. Our goal is that agent-based knowledge discovery will allow us to maximize the usage of distributed computing resources, and minimize the network traffic, as well as facilitate the easy integration and use of multiple learning algorithms. Our proposed architecture meets some principles, which are asserted in the literature are requirements for non-uniform, multiadministrator WAN environments:

• Scalability to a large number of cooperating sites: In a WAN environment, there may be a large number of sites which wish to share data.

• Data mobility: It should be easy and efficient to change the ŚŚhomeŠŠ of an object. Preferably, the object should remain available during movement.

• Act locally, think globally: agents on each site examine, analyze and learn from local data sources and integrate the knowledge they produce into a global result.

• Total local autonomy: Each site must have complete control over its own resources. This includes what objects to store and what actions agents have to perform to discover knowledge.

• Total local autonomy: Each site must have complete control over its own resources. This includes what objects to store and what actions agents have to perform to discover knowledge.

• Easily configurable policies: It should be easy for a local site to change the behavior of an agent if new better algorithms occur.

References

[1] Pearl, J. *Probabilistic reasoning in intelligent systems*, Morgan and Kaufman, San Mateo, CA, 1988

[2] G. F. Cooper, “The computation complexity of probabilistic inference using Bayesian belief networks,” *Artificial Intelligence*, Vol. 42, pp. 393Ű405, 1990.

[3] I.H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[4] F. G. Cozman, "Computing posterior upper expectations," *International Journal of Approximate Reasoning*, vol. 24, pp. 191-205, 2000.

[5] R.E. Neapolitan, *Probabilistic Reasoning in Expert Systems*, Wiley & Sons, Inc., 1990.

[6] "PROMEDAS,a probabilistic decision support system for medical diagnosis. " *Technical report, SNN Foundation for Neural Networks and University Medical Center Utrecht, Nijmegen, Netherlands, 2002. http://www.snn.kun.nl/ bert/diagnosis*

[7] G. Assmann, P. Cullen, and H. Schulte, "The Munster heart study (PROCAM): results of follow-up at 8 years. ," *Eur. Heart J.*, Vol. 19, pp. A2-A11, 1998.

[8] G. Assmann, P. Cullen, and H. Schulte, "Simple scoring scheme for calculating the risk of acute coronary events based on the 10-year follow-up of the Prospective Cardiovascular Munster (PROCAM) Study. ," *Circulation*, Vol. 105, pp. 310-315, 2002.

[9] S. Sian, "Extending Learning to Multiple Agents: Issues and a Model for Multi-Agent Machine Learning (MAML)," *Proceedings of the European Working Session on Learning (EWSL91)*, Y. Kodratroff (Ed.), Springer-Verlag, pp. 458-472, 1991.

[10] Y. Shoham, "Agent-Oriented Programming," *Technical Report STAN-CS-90-1335*, Stanford University, 1990.

[11] M. Wooldridge & N. R. Jennings, "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review*, Vol. 10, 1995.

[12] M. T., Oszu, P., Valduriez, *Principles of Distributed Database Systems*, Prentice Hall, Englewood Cliffs, NJ, 1999

[13] M. Piattini, and O. Diaz , *Advanced Database Technology and Design*, Artech House, Inc. 685 Canton Street Norwood, MA 02062, 2000.

Horea Adrian Grebla
Babes-Bolyai University Cluj-Napoca
Computer Science Department
Address: 1, Mihail Kogalniceanu St.,400084 Cluj-Napoca, Romania
E-mail: horea@cs.ubbcluj.ro

Calin Ovidiu Cenan
Technical University Cluj-Napoca
Computer Science Department
Address: 26-28, Baritiu St., 400027 Cluj-Napoca, Romania
E-mail: calin.cenan@cs.utcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 251-255

Data Acquisition Sistem for Vibration Signal Analysis

Florin Grofu, Luminita Popescu, Marian Popescu

Abstract: Antifriction bearing failure is a major factor in failure of rotating machinery. As a fatal defect is detected, it is common to shut down the machinery as soon as possible to avoid catastrophic damages. Performing such an action, which usually occurs at inconvenient times, typically results in substantial time and economical losses. It is, therefore, important to monitor the condition of antifriction bearings and to know the details of severity of defects before they cause serious catastrophic consequences. The vibration monitoring technique is suitable to analyze various defects in bearing. This technique can provide early information about progressing malfunctions. In this paper is presented a data acquisition system conceived for the analysis of the signal from two vibration transductor

Keywords: Antifriction bearings; Prediction; Vibration signal; Acquisition System.

1 Introduction

Condition monitoring of antifriction bearings in rotating machinery using vibration analysis is a very well established method. It offers the advantages of reducing down time and improving maintenance efficiency. The machine need not be stopped for diagnosis. Even new or geometrically perfect bearings may generate vibration due to contact forces, which exist between the various components of bearings. Antifriction bearing defects may be categorized as localized and distributed. The localized defects include cracks, pits and spalls caused by fatigue on rolling surfaces. The other category, distributed defects include surface roughness, waviness, misaligned races and off size rolling elements. These defects may result from manufacturing error and abrasive wear. Antifriction bearing failures result in serious problems, mainly in places where machines are rotating at constant and high speeds. To prevent any catastrophic consequences caused by a bearing failure, bearing condition monitoring techniques, such as, vibration analysis and acoustic emission analysis have been developed to identify existence of flaws in running bearings. Vibration signature monitoring and analysis in one of the main techniques used to predict and diagnose various defects in antifriction bearings. Vibration signature analysis provides early information about progressing malfunctions and forms the basic reference signature or base line signature for future monitoring purpose. Defective rolling elements in antifriction bearings generate vibration frequencies at rotational speed of each bearing component and rotational frequencies are related to the motion of rolling elements, cage and races. Initiation and progression of flaws on antifriction bearing generate specific and predictable characteristic of vibration. Components flaws (inner race, outer race and rolling elements) generate a specific defect frequencies calculated from equations, mentioned by Chaudhary and Tandon, namely:

Inner face malfunction frequency $f_i = \frac{n}{2} f_r \left[1 + \left(\frac{BD}{PD}\right) \cos\beta\right]$

Outer race malfunction frequency $f_o = \frac{n}{2} f_r \left[1 - \left(\frac{BD}{PD}\right) \cos\beta\right]$

Roller malfunction frequency $f_R = \frac{BD}{PD} f_r \left[1 - \left(\frac{BD}{PD}\right)^2 \cos^2\beta\right]$

Where:BD roller diameter, PD pitch diameter, *fr* rotational frequency, *n* number of rollers and ß angle of contact.The time domain and frequency domain analyses are widely accepted for detecting malfunctions in bearings. The frequency domain spectrum is more useful since it also identifies the exact nature of defect in the bearings.

2 Experimental Stand

An experimental test rig built to predict defects in antifriction bearings is shown in (figure 1). The test rig consists of a shaft with central rotor, which is supported on two bearings. An induction motor coupled by a flexible coupling drives the shaft. Self aligning double row ball bearing is mounted at driver end and cylindrical roller bearing is mounted at free end. The cylindrical roller bearing is tested at constant speed of 1400 rpm. Cylindrical roller bearing type 6308C3 (with outer race and roller defects and with inner race defect) have been used for analysis

The details of the bearings used in the present analysis are: number of roller 8, outer diameter 90mm, inner diameter 40mm, pitch diameter 55mm, roller diameter 15mm, contact angle β=0˚

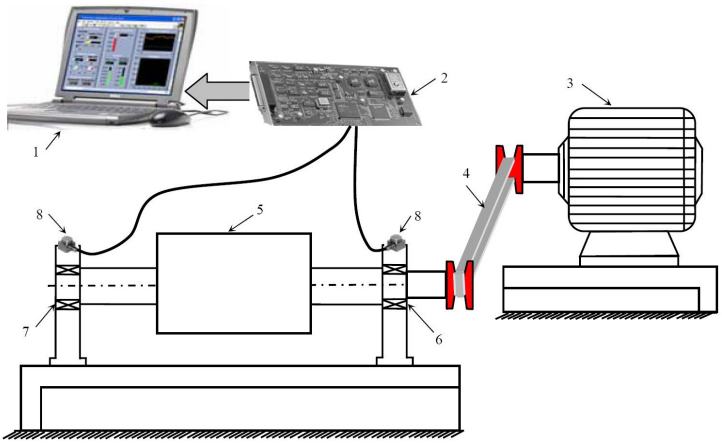


Figure 1: 1–PC; 2–Data Acquisition System; 3–Motor; 4–Flexible Coupling; 5–Rotor; 6–Self Aligning Ball Bearing; 7–Roller Bearing; 8–Transductors

In order to make the acquisition of the useful signal, a vibration transductor was put on each bearing. The signal from the two sensors is acquisitioned and transmitted towards a computer which will do the frequency analysis of the received signals. The block scheme of the acquisition part is presented in the following figure (figure 2).

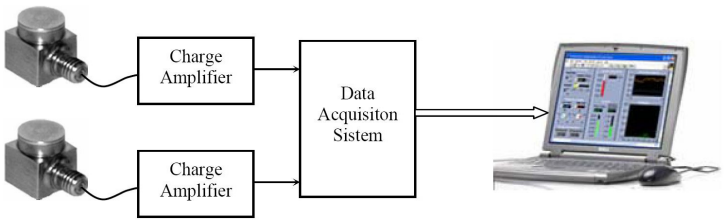


Figure 2:

2.1 Vibration transductor

The vibration transductor that is used is MAQ 36. Model MAQ36 charge output miniature accelerometer is designed to be used in Industrial test and automation environments; including laboratory testing, modal studies and test cells where high temperatures are likely to be encountered and where space is limited and small size is desired or a high natural frequency is required. The MAQ36 is a self generating piezoelectric transducer which has no internal electronics and requires no external power for operation. These units are usually connected to a local charge amplifier that is mounted as near as possible in a lower temperature environment. The seismic element is mechanically isolated from the mounting base, resulting in a low base strain sensitivity. The stainless steel materials are non-magnetic resulting in very low magnetic field susceptibility. These features, together with a sealed body, assure accurate and reliable data.

The accelerometer is the transducer type most commonly encountered when measuring vibration levels. The accelerometer normally consists of a seismic mass mechanically connected to the accelerometer base through a piezoelectric material. Piezoelectric materials have the property of producing electrical charge when bent and twisted (even shear forces will work here). Basically, a charge is generated. Hence, a charge sensitive amplifier will produce a signal independent of the capacitance of the cable between the accelerometer and the amplifier.

3 Acquisition system

To analyze the vibrations, the frequency analysis method is used, which implies the implementation on a PC of the Fourier transform. In order to have the most correct analysis, the acquisition rate of the useful signal must be as

big as it can reporting to the maximum frequency of the analyzed signal's spectrum. Making such an acquisition system implies to fulfill simultaneously more conditions which most of the times are in contradiction.

First, a very high speed of acquisition is required, which implies a big volume of data. To make a real time analysis of these data, there must be used complex systems with high power of calculus, like the process computers. The vibrations produced in the bearings which must be analyzed, appear in an improper industrial environment. Regarding the climate and mechanical emplacement, the usual process computers cannot be positioned next to the process.

The alternative would be placing an acquisition system next to the two sensors and transmitting the information in a numerical way to the computer, which is placed in the control room. In this case, a high speed numerical transmission is necessary, so that the analysis of the acquisitioned signals to be in real time. Generally, data transmission at high speeds can be made through parallel interfaces (with a big number of wires) or through core complicated serial interfaces.

Because the defects that appear in the bearings is a slow phenomenon, for the analysis of the defects using vibrations analysis, a fast signal acquisition can be made for a certain period of time with local hold of the data and then a slower transmission of the data through a standard serial interface (for example RS 485). The acquisition system must be able to make acquisitions of at least 100.000 samples/second and to hold a big number of samples (the equivalent of an interval of few seconds to observe the useful signal). The structure of the proposed acquisition system is presented in figure 3:

3.1 Core with microcontroller

The core with microcontroller is made with DS89C420. The DS89C420 offers the highest performance available in 8051-compatible microcontrollers. It features a redesigned processor core that executes every 8051 instruction (depending on the instruction type) up to 12 times faster than the original for the same crystal speed. Typical applications see a speed improvement of 10 times using the same code and crystal. The DS89C420 offers a maximum crystal speed of 33MHz, achieving execution rates up to 33 million instructions per second (MIPS).

DS89C420 is 8051 Pin- and Instruction-Set Compatible and have four Bidirectional I/O Ports,Three 16-Bit Timer Counters, 256 Bytes Scratchpad RAM, 16kB Flash Memory, In-System Programmable through Serial Port, Dynamically Adjustable by Software, 1 clock-per-machine cycle and Single-cycle instruction in 30ns, Optional variable length MOVX to access fast/slow peripherals, Dual data pointers with auto increment/decrement and toggle select, Programmable clock divider, Two full-duplex serial ports, Programmable watchdog timer, 13 interrupt sources (six external), Five levels of interrupt priority

3.2 RAM memory

The RAM memory used for the local hold of the conversions result is of the DS1270Y type.

The DS1270Y 16M Nonvolatile SRAMs are 16,777,216-bit, fully static nonvolatile SRAMs organized as 2,097,152 words by 8 bits and read and write access times as fast as 70 ns. Each NV SRAM has a self-contained lithium energy source and control circuitry which constantly monitors VCC for an out-of-tolerance condition. When such a condition occurs, the lithium energy source is automatically switched on and write protection is unconditionally enabled to prevent data corruption. There is no limit on the number of write cycles which can be executed and no additional support circuitry is required for microprocessor interfacing. Optional industrial temperature range of -40˚C to +85˚C, designated IND.

3.3 Analog-to-Digital Converter

The analog-to-digital converter which is used is MAX 120. The MAX 120 is BiCMOS, sampling 12-bit analog-to-digital converters (ADCs) combine an on-chip track/hold (T/H) and low-drift voltage reference with fast conversion speeds and low power consumption. The T/H's 350ns acquisition time combined with the 1.6μs conversion time results in throughput rates as high as 500k samples per second (ksps).The MAX 120 accepts analog input voltages from -5V to +5V and operates with clocks in the 0.1MHz to 8MHz frequency range.

The MAX 120 employ a standard microprocessor (μP) interface. Three-state data outputs are configured to operate with 12-bit data buses. Data-access and bus-release timing specifications are compatible with most popular μPs without resorting to wait states. In addition, the MAX 120 can interface directly to a first in, first out (FIFO) buffer, virtually eliminating μP interrupt overhead.

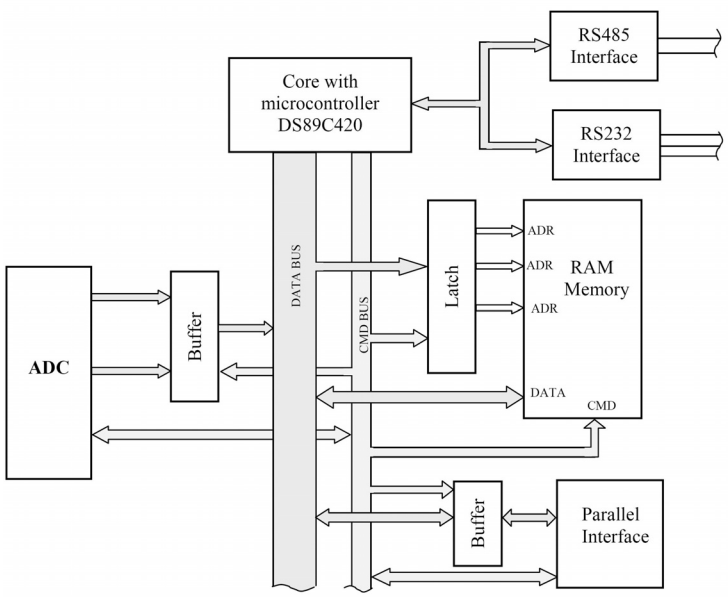


Figure 3:

The data transfer between the analog-to-digital converter and memory or between the memory and the parallel interface can be made through the microcontroller or directly on the data bus. In this case, the microcontroller generates only command signals which are necessary for data transfer. To increase the acquisition speed, saving previous data is done simultaneously with a new conversion. In this way, speeds of 400.000 samples/second can be reached. Transmitting data from the RAM memory towards the PC can be done serial (RS 485) for big distances, or serial (RS 232) or parallel on small distances.

4 Software

The acquisition program is made with the help of LabWindows/CVI 7.1 and is split into 3 components:

- one component which makes the acquisition and data process and which is conceived as a “service” under Windows XP.
- one MySQL component (data base server) which asynchronously takes over the files stored on the HDD by the first component and stores them into a data base.
- one component which is launched at demand and which presents the data in the data base into a suggestive graphical interface.

If the acquisition board is placed at a considerable distance from the PC, the acquisition is made through serial RS 232 by a RS 485/RS 232 converter. The program works on interrupt, meaning to receive data in the RS 232 serial buffer, and when it is full (or we can set a certain number of characters, smaller than the buffer’s dimension), an interrupt is generated. The function which deals with the interruption downloads the buffer of the serial into a memory variable of string type as in the example : unit data[100.000] where data[i] = [0x0F& high_octet] *256 + low_octet meaning high octet has the first n bits equal to zero. (the analog-to-digital converter works on 12 bits).

These data from the memory are processed in the following way:

- applying the Fourier transform, will result the frequency spectrum of the data from the memory.
- from the frequency spectrum analysis, a diagnosis of the device try is made, meaning the type of defect (for example, for the bearings: broken ball, deficient oiling, usage, the engine and pump are not on the center).
- the conclusions from the previous step are being transmitted to the dispatcher for a real time information on the defects. If the found defects are major, the “emergency stop” of the device is necessary.

- the frequency characteristic from step 1 is memorized in the computer's memory.

According to the diagnosis algorithms, there may be certain frequency domains which must be analyzed normally, thing that can be done with performant digital filters. If defects are found, the entire acquisitioned frequency spectrum is analyzed. This is done not to have high time values for processing or to eliminate parasite signals not related to the process.

These times for acquisition, processing, memorizing, must be included into the sampling period chosen for the process if the same computer coordinates the whole process.With the help of the memorized files, a history of the functioning can be made, for two purposes:

- visualizing in time the evolution of the device, useful for revision and repairs planning.
- creating and improving some algorithms for defects identification corresponding to the analyzed physic device.

5 Summary and Conclusions

Time waveform and frequency spectrum provide useful information to analyze defects in antifriction bearings. Time waveform indicates severity of vibration in defective bearings. Frequency domain spectrum identifies amplitudes corresponding to defect frequencies and enables to predict presence of defects on inner race, outer race and rollers of antifriction bearings. The distinct and different behaviour of vibration signals from bearings with inner race defect, outer race defect and roller defect helps in identifying the defects in roller bearings.

Also, the acquisition system presented ensures the obtaining of some precise acquisitions which correspond from the point of view of resolution (number of bits) and of speed of acquisition. It's simple and robust structure allows this system to be made according to any industrial environment.

References

[1] Y Li and C Zhang. 'Dynamic Prognostic Prediction of Defect Propagation on Rolling Element Bearing'. *Journal of Vibration and Acoustics, Trans of ASME*, vol 120, no 1, pp 214-220.

[2] Teruo Igarishi and Hiroyoshi. 'Studies on Vibration and Sound of Defective Rolling Bearings'. *Bulletin JSME*, vol 25, no 204, 1980, pp 994-1001.

[3] I J Taylor. 'Identification of Bearing Defects by Spectral Analysis'. *Journal of Mechanical Design, Transaction of ASME*, vol 120, 1980, pp 199-204.

[4] Porat B., *A Course in Digital Signal Processings*, John Wiley & Sons, Inc., New York, 1997.

[5] D. H. Sheingold, *Analog-Digital Conversion Handbook*, Analog Devices, Inc.

[6] J Jacob Wikner, *STUDIES ON CMOS DIGITAL-TO-ANALOG CONVERTERS*, Linköping Studies in Science and Technology Dissertation No. 667, Sweden Linköping 2001.

[7] Maxim , *Digital-Analog Converters Are a "Bit" Analog*, Application Note 1055: Apr 16, 2002.

Grofu Florin, Luminita Popescu, Marian Popescu
Constantin Brancusi" University of Tg-Jiu
Department of Automation
Address: 2, Geneva, Tg-Jiu, Romania
E-mail: florin@utgjiu.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 256-261

Descriptive Timed Membrane Petri Nets for Modelling of Parallel Computing

Emilian Guţuleac

Abstract: In order to capture the compartmentation and behaviour of membrane systems for modelling of parallel computing, we introduce the descriptive dynamic rewriting Descriptive Membrane Timed Petri Nets (DM-nets) that can at in run-time modify their own structure by rewriting some of their descriptive expression components. Furthermore, this descriptive approach facilitates the understanding of complex models and their component-based construction as well as the application of modern computer engineering concepts.

Keywords: Descriptive Petri nets, membrane systems, modelling, parallel computing.

1 Introduction

Recent technological achievements require advances beyond the existing computational models in order to be used effectively. Pragmatic aspects of current and future computer systems will be modelled so that realistic estimates of efficiency can be given for algorithms in these new settings.

Petri nets (PN) are very popular formalism for the analysis and representation of parallel and distributed computing in concurrent systems that has draw much attention to modelling and verification of this type of systems [1].

P systems, also referred to as membrane systems, are a class of parallel and distributed computing models [6]. The interest of relating P systems with the PN model of computation lead to several important results on simulation and decidability issues. Some efforts have been made to simulate P systems with Petri nets [2, 5, 7] to verifying the many useful behavioral properties such as reachability, boundedness, liveness, terminating, etc.

In this paper we propose a new approach to express the components of continuous-time P systems [6] throughout components of escriptive Petri Nets (PN) using descriptive expressions (DE) [3] for modelling of parallel computing. The DE are used for analytical representation and compositional construction of PN models. To model specific rules of P-systems within the framework of the descriptive Rewriting Timed PN (RTN) [4] we introduce a new extensions Ű the descriptive Membrane RTN, called DM-nets, that can modify dynamically their own structures by rewriting rules some of their components.

2 Labeled Extended Petri Nets

In this section, we define a variant of PN called labeled extended PN. Let L be a set of labels $L = L_P \uplus L_T$. Each place p_i labeled $l(p_i) \in P$ a local state and transition t_j has action labeled as $l(t_j) \in L_T$.

A labeled extended PN is structure as a $\Gamma =< P, T, Pre, Post, Test, Inh, G, Pri, K_p, l >$, where: P is the finite set of places and T is a finite set of transitions that $P \cap T = \emptyset$. In the graphical representation, the place is drawn as a circle and the transition is drawn as a black bar; The *Pre*, *Test* and $Inh : P \times T \times \mathcal{N}^{|\mathcal{P}|} \to \mathcal{N}_+$ respectively is a forward flow, test and inhibition functions and is a backward flow function in the multi-sets of P, where defined the set of arcs A and describes the marking-dependent cardinality of arcs connecting transitions and places. The set A is partitioned into tree subsets: A_d, A_h, and A_t. The subset A_d contains the directed arcs which can be seen as A_d: $((P \times T) \cup (T \times P)) \times \mathcal{N}^{|\mathcal{P}|} \to \mathcal{N}_+$ and are drawn as single arrows. The inhibitory arcs $A_h : (P \times T) \times \mathcal{N}^{|\mathcal{P}|} \to \mathcal{N}_+$are drawn with a small circle at the end. The test arcs $A_t : (P \times T) \times \mathcal{N}^{|\mathcal{P}|} \to \mathcal{N}_+$are directed from a place to a transition, and are drawn as dotted single arrows. It does not consume the content of the source place. The arc of net is drawn if the cardinality is not identically zero and this is labeled next to the arc and by a default value being 1; $G : E \times \mathcal{N}^{|\mathcal{P}|} \to \{true, false\}$ is the guard function transitions. For $t \in T$ a guard function $g(t,M)$ that will be evaluated in each marking, and if it evaluates to *true*, the transition t may be enabled, otherwise t is disabled (the default value is *true*); $Pri : T \to \mathcal{N}_+$ defines the priority functions for the firing of each transition that maps transitions onto natural numbers representing their priority level. The enabling of a transition with higher priority disables all the lower priority transitions; $K_p : P \to \mathcal{N}_+$ is the capacity of places, and by default being infinite value; The $l : T \cup P \to L$, is a labeling function that assigns a label to a transition and places. In this way that maps transition name into action names that $l(t_j) = l(t_k) = \alpha$ but $t_j \neq t_k$ and $l(p_i) = l(p_n) = \beta$ but $p_i \neq p_n$.

A marked labeled extended PN net is a pair $N =< \Gamma, M_0 >$, where Γ is a labeled PN structure and M_0 is the initial marking of the net. $M : P \to \mathcal{N}_+$ is the current marking of net which is described by a symbolic vector-column $M = (m_i p_i), m_i \geq 0, \forall p_i \in P$, where the $(m_i p_i)$ is the number m_i of tokens in place p_i. The M is the state of net that assigns to each place tokens, represented by black dots.

The details concerning on enabling and firing rules, and evolution for of $N =< \Gamma, M_0 >$ can be found in [3] as they require a great deal of space.

3 Descriptive expressions of Petri nets

Due to the space restrictions we will only give a brief overview to this topic and refer the reader to [3, 4] and the references therein. In following for abuse of notation, labels and name of transitions/places are the same. We use the concept of a basic descriptive element (*bDE*) for a basic PN (*bPN*) introduced in [2] as following: $bDE = |_{t_j}^{\alpha_j} m_i^0 p_i[W_i^+, W_i^-]|_{t_k}^{\alpha_k}$. The translation of this *bPN* is shown in figure 1a, where respectively is input transition (action type α_j) and $t_k = p_i^\bullet$ is the output transition (action type α_k) of place $p_i \in P$ with initial marking m_i^0, and the flow type relation functions $W_i^+ = Pre(t_j, p_i)$ and $W_i^- = Post(t_j, p_i)$, respectively which return the multiplicity of input and output arcs of the place $p_i \in P$. The derivative elements of *bDE* are for $p_i^\bullet = \emptyset, W_i^- = 0$ is $|_{t_j}^{\alpha_j} m_i^0[W_i]$ with final place p_i of t_j and ${}^\bullet p_i = \emptyset, W_i^+ = 0$ is $m_i^0 p_i[W_i]|_{t_k}^{\alpha_k}$ with entry place p_i of t_k. If the initial marking m_i^0 of place is a zero tokens we can omit m_i^0 in *bDE*. By default, if the type of action α is not mentioned this to match the name of a transition t. From a *bDE* we can build more complex DE of PN components by using composition operations. Also by default,if $W_i^+ = W_i^- = 1$, we present *bDE* and it derivatives as following:$|_{t_j}^{\alpha_j} m_i^0 p_i|_{t_k}^{\alpha_k}$, $|_{t_j}^{\alpha_j} m^0 p_i$ or $m_i^0 p_i|_{t_k}^{\alpha_k}$.

A descriptive expression (*DE*) of a labeled PN is either *bDE* or a composition of *DE* a *N*: $DE ::= bDE | DE * DE | \circ DE$, where $*$ represents any binary composition operation and $\circ$ any unary operation.

Descriptive Compositional Operations. In the following by default the labels of *N* are encoded in the name of the transitions and places. The composition operations are reflected at the level of the *DE* components of *N* models by fusion of places, fusion of transitions with same type and same name (label) or sharing of as subnets.

Place-Sequential Operation. This binary operation, denoted by the " | " *sequential operator*, determines the logic of a interaction between two local states p_i (pre-condition) and p_k (post-condition) by t_j action that are in precedence and succeeding (causality-consequence) relation relative of this action. Sequential operator is the *basic mechanism* to build *DE* of *N* models. This operation is an *associative*, *reflexive* and *transitive* property, but is *not commutative* operation. The means the fact $DE1 = m_i^0 p_i[W_i]|_{t_j}^{\alpha_j} m_k^0 p_k[W_k] \neq m_k^0 p_k[W_k]|_{t_j}^{\alpha_j} m_i^0 p_i[W_i]$ that the specified conditions (local state) associated with place-symbol p_i are fulfilled always happens before then the occurrence of the conditions associated with place-symbol p_k by means of the action t_j. Also, the PN modelling of the *iteration* operation is obtained by the fusion of head (entry) place with the tail (final) place that are the same name (*closing* operation) in *DE* which describes this net. The self-loop of *N*2 net described by an: $DE2 = m_i^0 p_i[W_i]|_{t_j}^{\alpha_j} p_i[W_i] = m_i^0 \tilde{p}_i[W_i]|_{t_j}^{\alpha_j}$, it is the test operator "$\tilde{p}$", i.e. represent the *test* arc. The translation of $DE2$ in *N*2 is shows in figure 2b.

Inhibition Operation. This unary operation is represented by inhibitory operator "- " (place-symbol with over-bar) and it $DE3 = m_i^0 \overline{p_i}[W_i]|_{t_j}^{\alpha_j}$ describe the inhibitor arc with a weight $W_i = Inh(p_i, t_j)$.

Synchronization Operation. This binary operation is represented by the " $\bullet$" or " $\wedge$"*join* operator describe the rendez-vous synchronization (by transition t_{t_j}) of a two or more conditions represented respectively by symbol-place $p_i \in {}^\bullet t_j, i = \overline{1,n}$, i.e. it indicate that all preceding conditions of occurrence actions must have been completed. This operation is a commutative, associative and reflexive.

Split Operation . This binary operation represented by the " $\diamondsuit$" *split* operator and it describe the causal relations between activity $t_j$and its post-conditions: after completion of the preceding action of t_j concomitantly several other post-condition can take occurs in parallel ("message sending"). Property of split operation is a commutative, associative and reflexive.

Competing Parallelism Operation. This compositional binary operation is represented by the " $\vee$" competing parallelism operator, and it can be applied over two N_A with $DE_A = A$ and N_B with $DE_B = B$ or internally into resulting N_R with $DE_R = R$, between the places of a single N_R which the symbol-places with the same name are fused, respectively. We can represent the resulting $DE_R = A \vee B$ as a set of ordered pairs of places with the same name to be fused, with the first element belonging to *A* the second to *B*. The fused places will inherit the arcs of the place in *A* and *B* . Also, this compositional binary operation is a *commutative*, *associative* and *reflexive* property.

Precedence Relations between the Operations. We introduce the following precedence relation between the compositional operations in the *DE*: a) the evaluation of operations in DE are applied left-to-right; b) an unary operation binds stronger than a binary one; c) the "• "operation is superior to"|" and " ♢", in turn, its are superior the "∨ " operation. Further details on definitions, enabling and firing rules, and evolution for of N can be found in [3] as they require a great deal of space.

4 Dynamic Rewriting Petri Nets

In this section we introduce the model of *descriptive dynamic net rewriting* PN system. Let $X\rho Y$ is a binary relation. The *domain* of is the $Dom(\rho) = \rho Y$and the *codomain* of ρ is the $Cod(\rho) = X\rho$. Let $A =< Pre, Post, Test, Inh >$ is a set of arcs belong to netΓ .

A descriptive dynamic rewriting *PN* system is a structure $RN =< \Gamma, R, \phi, G_{tr}, G_r, M >$,
where: $=< P, T, Pre, Post, Test, Inh, G, Pri, K_p, l >$; $R = r_1, ..., r_k$ is a finite set of rewriting rules about the runtime structural modification of net that $P \cap T \cap R = \emptyset$. In the graphical representation, the rewriting rule is drawn as a two embedded empty rectangle. We let $E = T \cup R$ denote the set of events of the net; $\phi : E \to T, R$ is a function indicate for every rewriting rule the type of event can occur; $G_{tr} : R \times \mathbb{N}^{|\mathcal{P}|} \to \{true, false\}$ and $G_r : R \times \mathbb{N}^{|\mathcal{P}|} \to \{true, false\}$ is the *transition rule guard function* associated with $r \in R$ and the rewriting rule guard function defined for each rule of $r \in R$, respectively. For $\forall r \in R$, the $g_{tr} \in G_{tr}$ and $g_r \in G_r$will be evaluated in each marking and if its are evaluates to *true*, the rewriting rule *r* may be *enabled*, otherwise it is disabled. Default value of $g_{tr} \in G_{tr}$ is *true* and for $g_r \in G_r$ is *false*. Let $RN =< R\Gamma, M >$ and $R\Gamma =< \Gamma, R, \phi, G_{tr}, G_r >$ described with the descriptive expression $DE_{R\Gamma}$ and DE_{RN}, respectively. A dynamic rewriting structure modifying rule $r \in R$ of *RN* is a map $r : DE_L \rhd DE_W$, where whose *codomain* of the rewriting operator $\rhd$ is a fixed descriptive expression DE_L of a subnet RN_L of current net *RN*, where $RN_L \subseteq RN$,with $P_L \subseteq P$, $E_L \subseteq E$and set of arcs $A_L \subseteq A$ and whose *domain* of the $\rhd$ is a descriptive expression DE_W of a new RN_W subnet with $P_W \subseteq P$, $E_W \subseteq E$ and set of arcs A_W. The $\rhd$ rewriting operator represent binary operation which produce a *structure change* in the DE_{RN} and the net *RN* by replacing (rewriting) of the fixed current DE_L of subnet RN_L (DE_L and RN_L are dissolved) by the new DE_W of subnet RN_W now belong to the new modified resulting $DE_{RN'}$ of net $RN^{'} = (RN \backslash RN_L) \cup RN_W$ with $P^{'} = (P \backslash P_L) \cup P_W$ and $E^{'} = (E \backslash E_L) \cup E_W$, where $A^{'} = (P \backslash P_L) \cup A_W$ the meaning of $\backslash$ (and $\cup$) is operation to removing (adding) RN_L from ($RN_W$to) net *RN*. In this new net $RN^{'}$, obtained by execution (fires) of enabled rewriting rule $r \in R$, the places and events with the same attributes which belong $RN^{'}$ are fused, respectively. By default the rewriting rules $r : DE_L \rhd \emptyset$ and$r : \emptyset \rhd DE_W$ describe the rewriting rule which fooling holds $RN^{'} = (RN \backslash RN_L)$ and $RN^{'} = (RN \cup RN_W)$, respectively. A state of a net *RN* is a pair $(R\Gamma, M)$, where $R\Gamma$ is the configuration of net together with a current marking M. Also, the pair $(R\Gamma_0, M_0)$ with $P_0 \subseteq P$, $E_0 \subseteq E$ and marking M_0 is called the initial state of the net.

Enabling and Firing of Events. The enabling of events depends on the marking of all places. We say that a transition t_j of event e_j is enabled in current marking M if the following enabling condition $ec(t_j, M)$ is verified:

$ec(t_j, M) = (\wedge_{\forall p_i \in {}^\bullet t_j}(m_i \geq Pre(p_i, t_j)) \wedge (\wedge_{\forall p_k \in {}^\circ t_j}(m_k < Inh(p_i, t_j))) \wedge (\wedge_{\forall p_l \in {}^* t_j}(m_l \geq Test(p_l, t_j))) \wedge (\wedge_{\forall p_n \in t_j^\bullet}((K_{p_n} - m_i) \geq Post(p_n, t_j)))) \wedge g(t_j, M))$.

Similarly, the rewriting rule $r_j \in R$ is enabled in current marking M if the following enabling condition $ec_{tr}(r_j, M)$is verified:

$ec_{tr}(r_j, M) = (\wedge_{\forall p_i \in {}^\bullet r_j}(m_i \geq Pre(p_i, r_j)) \wedge (\wedge_{\forall p_k \in {}^\circ r_j}(m_k < Inh(p_i, r_j))) \wedge (\wedge_{\forall p_l \in {}^* r_j}(m_l \geq Test(p_l, r_j))) \wedge (\wedge_{\forall p_n \in r_j^\bullet}((K_{p_n} - m_i) \geq Post(p_n, r_j)))) \wedge g(r_j, M))$.

Let the $T(M)$ and $R(M)$ is respectively the set of enabled transitions and rewriting rule in current marking M. Let the $E(M) = T(M) \uplus R(M)$, is the set of enabled events in a current marking M. The event $e_j \in E(M)$ fire if no other event $e_k \in E(M)$ with higher priority has enabled. Hence, for e_j event $if((\phi_j = t_j) \vee (\phi_j = r_j) \wedge (g_{tr}(r_j, M) = false))$ *then* (the firing of transition $t_j \in T(M)$ or rewriting rule $r_j \in R(M)$ change only the current marking:$(R\Gamma, M) \xrightarrow{e_j} (R\Gamma, M^{'}) \Leftrightarrow (R\Gamma = R\Gamma^{'}$ and $M[e_j > M^{'})$. Also, for e_j event $if((\phi_j = r_j) \wedge (g_r(r_j, M) = true)$ *then* (the event e_j occur to firing of rewriting rule r_j and it occurrence change configuration and marking of current net:$(R\Gamma, M) \xrightarrow{r_j} (R\Gamma^{'}, M^{'})$, $M[r_j > M^{'})$.

The accessible state graph of a net$RN =< \Gamma, M >$ is the labeled directed graph whose nodes are the states and whose arcs which is labeled with events of *RN* are of two kinds: a) firing of a enabled event $e_j \in E(M)$: arcs from state $(R\Gamma, M)$ to state$(R\Gamma, M^{'})$ labeled with evente_j then this event can fire in the net configuration $R\Gamma$ at marking M and leads to new marking $M^{'}$: $(R\Gamma, M) \xrightarrow{e_j} (R\Gamma^{'}, M^{'}) \Leftrightarrow (R\Gamma = R\Gamma^{'}$ and $M[e_j > M^{'}$ in $R\Gamma)$; b) change configuration:

arcs from state $(R\Gamma, M)$ to state($R\Gamma', M'$) labeled with rewriting rule r_j :$(R\Gamma_L, M_L) \triangleright (R\Gamma_W, M_W)$ which represent the change configuration of current RN net: $(R\Gamma, M) \xrightarrow{e_j} (R\Gamma', M')$ and $M[r_j > M'$.

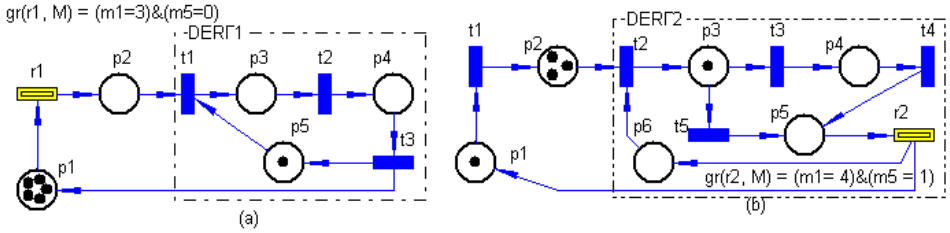


Figure 1: Translation of (a) $DE_{R\Gamma 1}$in $RN1$ and (b) $DE_{R\Gamma 2}$ in $RN2$

Let we consider the $RN1$ given by the following descriptive expression:$DE_{R\Gamma 1} = p_1|_{r_1} p_2 \vee DE'_{R\Gamma 1}$, $DE'_{R\Gamma 1} = (p_2 \cdot p_5)|_{t_1} p_3|_{t_2} p_4|_{t_3} (p_1 \diamondsuit p_5)$, $M_0 = (5p_1, 1p_5)$, $g_r(r_1, M) = (m_1 = 3)\&(m_5 = 0)$ and $r_1 : DE_{R\Gamma 1} \triangleright DE_{R\Gamma 2}$. Also, for r_j is required to identify if RN_L belong the $R\Gamma$. Upon firing, the enabled events or rewriting rule modify the current marking and/or and modify the structure and current marking of net $RN1$ in $RN2$ given by: $DE_{R\Gamma 2} = p_1|_{t_1} p_2 \vee DE'_{R\Gamma 2}$, $DE'_{R\Gamma 2} = (p_2 \cdot p_6)|_{t_2} p_3 (|_{t_3} p_4|_{t_4} p_5 \vee |_{t_5} p_5|_{r_2} (p_1 \diamondsuit p_6))$, $M = (1p_1, 3p_2, 1p_3)$, $g_r(r_2, M) = (m_1 = 4)\&(m_5 = 1)$, $r_2 = r_1^{-1} : DE_{R\Gamma 2} \triangleright DE_{R\Gamma 1}$.

Figure 1 show the translation of $DE_{R\Gamma 1}$ in $RN1$ and $DE_{R\Gamma 2}$ in $RN2$, respectively.

5 Dynamic Rewriting Timed Petri Nets

Systems are described in timed PN (TPN) as interactions of components that can performed a set of activities associated with events. An event $e = (\alpha, \theta)$, where $\alpha \in E$ is the type of the activity (action name), and θ is the firing delay.

A descriptive dynamic rewriting TPN as a $RTN =< RN, \theta >$, where: $RN =< \Gamma, R, \phi, G_{tr}, G_r, M >$, $\Gamma =< P, T, Pre, Post, Test, Inh, G, Pri, Kp, l >$ (see Definition 2 and 3) with set of events E which can be partitioned into a set E_0 of *immediate* events and a set E_τ of *timed* events $E = E_0 \uplus E_\tau$. The immediate event is drawn as a thin bar and timed event is drawn as a black rectangle for transition or a two embedded empty rectangle for rewriting rules, and $Pri(E_0) > Pri(E_\tau)$; $\theta : E \times \mathbb{N}^{|\mathcal{P}|} \to \mathcal{R}_+$ is the weight function that maps events onto real numbers $\mathcal{R}_+$ (delays or weight speeds). Its can be marking dependent. The delays $\theta(e_k, M) = d_k(M)$defining the events firing parameters governing its duration for each timed events of E_τ. If several timed events are enabled concurrently $e_j \in E(M)$ for $e_j \in {}^\bullet p_i = \forall e_j \in E : Pre(p_i, e_j) > 0$, either in competition or independently, we assume that a race race competition condition exists between them. The evolution of the model will determine whether the other timed events have been aborted or simply interrupted by the resulting state change. The $\theta(e_j, M) = w_j(M)$ is weight speeds of immediate events $e_{j \in E_0}$. If several enabled immediate events are scheduled to fire at the same time in *vanishing* marking M with the weight speeds, and the probability to enabled immediate event e_j can *fire* is: $q_i(M) = w(e_j, M) / \sum_{e_l \in (E(M) \& {}^\bullet p_i)} w(e_j, M)$,where $E(M)$ is the set of enabled events in M. An immediate events $e_j \in T_0$ has a zero firing time.

6 P Systems and Descriptive Timed Membrane Petri Nets

Here we give a brief review of P systems and its encoding with DM-nets. The main components of P systems are membrane structures consisting of membranes hierarchically embedded in the outermost skin membrane. A full guide for P systems can be referred to [3]. In general, a basic evolution-communication P system with *active membranes* (of degree $n \geq 0$) is $\Pi = (O, H, \mu, \Omega, (\rho, \pi))$, where: O is the alphabets of objects; H is a finite set of labels for membranes;μ is a membrane structure consisting of n membranes labeled with elements h in H;Ω is the configuration, that is a mapping from membranes of Π(nodes in μ) to multisets of objects $\omega_k \in \Omega, k = 1, , |\Omega|$, from O;ρ and π is respectively the set off developmental rules ρ_h and $\pi_h$its priorities , $h = 0, 1, , n-1$. Thus the can be of two forms of rules: a) the *object rules* (OR), i.e., evolving and communication rules concerning the objects; b) the membranes rules (MR), i.e., the rules about the structural modification of membranes.

Here we define DM-Nets for encoding of P systems mentioned above into descriptive dynamic rewriting TPN as a RTN. The basis for DM-Nets is a membrane RTN that is DE net structure comprise: places; transitions;

weighed directed arcs from places to transitions and vice-versa; a capacity for each place; weighed inhibitory and test arcs; priority and guard function of transitions.

The $DM-nets$ of degree $n \geq 0$ is a construct $DM = \vee_{h=0}^{n-1}[_h DE_h]_h$, where DE_h is the descriptive expression of RTN_h that represent the configuration of membrane $[_h\ ]_h$ in a P system Π.

Consider the P system Π.The encoding of Π into RTN_Π is decomposed into two separate steps. First, for every membrane $[_h\ ]_h$ we associate: to each object $\omega_i \in \Omega$ one place $p_{h,i} = [_h m_i^0 p_i]_h$ labeled as ω_i with the initial marking m_i^0, and to each rule $\rho_{h,j} \in \rho$ one event $e_{h,j} = [_h e_j]_h$ labeled as $\rho_{h,j}$ that acts on the this membrane. Second, for every membrane $[_h\ ]_h$ we define the DE_h of RTN_h that it correspond to the initial configuration of the P system Π as $[_h DE_h]_h$.

Let u,v , and $u_{'},v_{'}$, is a multiset of objects. The *evolving* object rule $\rho_{h',j}\colon [_h[_{h'}u \to v]_{h'}]_h$ with multiset of objects u,v , which will be kept in membrane $[_h]_h$ is encoded as $[_h[_{h'}p_u|_{t_j}p_v]_{h'}]_h$. The antiport rule $\rho_{h',j}\colon [_h u[_{h'}v]_{h'}]_h \to [_h v^{'}[_{h'}u^{'}]_{h'}]_h$, that realize a synchronized wich object c the exchange of objects, is encoded as $[_h[_{h'}(p_u \cdot p_v \cdot \tilde{p}_c)|_{t_j}(p_{u'} \diamondsuit p_{v'})]_{h'}]_h$. Also, the symport rule $\rho_{h',k}\colon [_h u[_{h'}]_{h'}]_h \to [_h[_{h'}u^{'}]_{h'}]_h$ that move objects from inside to outside a membrane, or vice-versa is encoded as $[_h[_{h'}(p_u \cdot \tilde{p}_c)|_{t_k}p_{u'}]_{h'}]_h$.

Because a configuration mean both a membrane structure and the associated multisets, we need rules for processing membranes and multisets of objects as:

$MR = Change, Dissolve, Create, Divide, Merge, Separate, Move.$

The above membrane rewriting rules (realized by the rewriting events in DE) are defined as follows:

*Change*rewriting rule $[_h[_{h'}(DE_{h'}, M_{h'})]_{h'}]_h \rhd [_h[_{h'}(DE_{h'}^{'}, M_{h'}^{'})]_{h'}]_h$ that in runtime the current structure and the multisets of objects to membrane h, encoded by descriptive expression $DE_{h,}$ and marking $M_{h,}$is changed in a new structure $DE_{h,}^{'}$ with new marking $M_{h,}^{'}$;

Dissolve rewriting rule $[_h(DE_h, M_h)[_{h'}(DE_{h'}, M_{h'})]_{h'}]_h \rhd [_h(DE_h, M_h^{'})]_h$ that the objects and sub-membranes of membrane $h^{'}$now belong to its parent membrane h , the skin membrane cannot be dissolved;

Create rewriting rule $[_h(DE_h, M_h)]_h \rhd [_h(DE_h^{'}, M_h^{'})[_{h'}(DE_{h'}^{''}, M_{h'}^{''})]_{h'}]_h$ with $M_h = M_h^{'} + M_{h'}^{''}$ that the new membrane $h^{'}$ is created and $M_{h'}^{''}$are added into membrane $h^{'}$, the rest remain in the parent membrane h; Divide rewriting rule $[_h(DE_h, M_h)]_h \rhd [_h[_{h'}(DE_h, M_h)]_{h'}[_{h''}(DE_h, M_h)]_{h''}]_h$that the objects and sub- membranes are reproduced and added into membrane $h^{'}$ and membrane $h^{''}$, respectively;

Merge rewriting rule that the objects of membrane $h^{'}$ and $h^{''}$ are added to a new membrane h is:
$[_h[_{h'}(DE_{h'}^{'}, M_{h'})]_{h'}[_{h''}(DE_{h''}^{''}, M_{h''}^{''})]_{h''}]_h \rhd [_h(DE_{h'}^{'} \vee DE_{h''}^{''}, M_{h'} + M_{h''}^{''})]_h$;

Separate rewriting rule is the counterpart of *Merge* is done by a rewriting rule of the form $\rhd [_h(DE_{h'}^{'} \vee DE_{h''}^{''}, M_{h'} + M_{h''}^{''})]_h \rhd [_h[_{h'}(DE_{h'}^{'}, M_{h'})]_{h'}[_{h''}(DE_{h''}^{''}, M_{h''}^{''})]_{h''}]_h$ with the meaning that the content of membrane h is split into two membranes, with labels $h^{'}$ and $h^{''}$.

*Move*rewriting rule where a membrane $h^{''}$ can be moved out or moved into a membrane $h^{'}$ as a whole is:
$[_h[_{h'}(DE_{h'}, M_{h'})[_{h''}(DE_{h''}^{''}, M_{h''}^{''})]_{h''}]_{h'}]_h \rhd [_h[_{h'}(DE_{h'}, M_{h'})]_{h'}[_{h''}(DE_{h''}^{''}, M_{h''}^{''})]_{h''}]_h$ or

$[_h[_{h'}(DE_{h'}, M_{h'})]_{h'}[_{h''}(DE_{h''}^{''}, M_{h''}^{''})]_{h''}]_h \rhd [_h[_{h'}(DE_{h'}, M_{h'})[_{h''}(DE_{h''}^{''}, M_{h''}^{''})]_{h''}]_{h'}]_h$.

Thus, using the $DM-Nets$ facilitates a compact and flexible specification to visual simulate of P systems with dynamic rewriting TPN nets that permit the verification of the its many useful behavioral properties such as reachability, boundedness, liveness, terminating, etc., and the performance evaluation of parallel computing models.

7 Summary and Conclusions

In this paper we have proposed an approach to the performance modeling of the behaviour of P-systems through a class of Petri nets, called Descriptive Membrane Timed PN (DM-nets). Based upon the introduction of a set of descriptive composition operation and rewriting rules attached with transitions for the creation of dynamic rewriting TPN, the membrane structure can be successfully encoded as a membrane descriptive rewriting timed Petri nets models which permit the description the behavioral state based process run-time structure change of P systems. We are currently developing a software visual simulator with a friendly interface for verifying and performance evaluation of descriptive rewriting TPN models and DM-nets.

References

[1] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Francheschinis, "Modeling with Generalized Stochastic Petri Nets," *ser. In Parallel Computing* ,New York: Wiley, 1995.

[2] S. Dal Zilio, E. Formenti, "On the Dynamics of PB System: a Petri Net View," *In Proceedings WMC 2003, Lecture Notes in Computer Science 2933* , Springer-Verlag, pp. 153-167, 2004.

[3] E. Gutuleac, "Descriptive Compositional Construction of GSPN Models for Performance Evaluation of Computer Systems," *In Proceedings of the 8-th International Symposium on Automatic Control and Computer Science, SACCS2004, 22-23 October*,Iasi, Romania, CD, 2004.

[4] E. Gutuleac, "Descriptive Dynamic Rewriting GSPN-based Performance Modeling of Computer Systems," *Proceedings of the 15th International Conference on Control Systems and Computer Science, CSCS15, 25-27 May 2005*,Bucuresti, Romania, pp. 656-661, 2005.

[5] J. Kleijn, M. Koutny, G. Rozenberg, "Towards a Petri Net Semantics for Membrane Systems," *In Proceedings of the WMC6 2005, July 18-21*, Wien, Austria, pp. 439-459, 2005.

[6] Gh. Paun, "Membrane Computing. An Introduction," *Natural computing Series. ed. G. Rozenberg, Th. Back, A.E. EibenJ.N. Kok, H.P. Spaink, Leiden Center for Natural Computing*, Springer-Verlag, Berlin, p. 420, 2002.

[7] Z. Qi, J. You, and H. Mao, "P Systems and Petri Nets," *Proceedings WMC 2003, Lecture Notes in Computer Science, vol. 2933,* Springer-Verlag, Berlin, pp. 387-403, 2003.

Emilian Guţuleac,
Technical University of Moldova,
Computer Science Department,
Address: 168, Bd. Stefan cel Mare, MD-2004,
Chişinău, Republic of Moldova
E-mail: egutuleac@mail.utm.md

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 262-267

Deriving DNA Public Keys from Blood Analysis

Tatiana Hodorogea, Mircea-Florin Vaida

Abstract: In this work we consider a technical process for protecting medical information and other data assets using a technique of deriving DNA public keys from blood analysis. A DNA encryption technique is further developed here in which a personŠs medical data is encrypted in DNA strands based on the central dogma of molecular biology. Protection is enhanced by using the patient's own blood mineral levels as a seed for selecting, transmitting, and recovering that personŠs public key.

Keywords: Security, DNA Public Keys, Cryptography, Genetic code

1 Introduction

Why we need security? That question involves many elements or we may consider only a technical process to increase the life security? A strong connection can be established between the material and spiritual life based on information, [17]. The topic of information security is very broad Ű involving issues ranging from physical protection of computer infrastructures to maintaining the privacy of individuals. As our society progresses into the information age, everyone from citizens to business and governmental institutions is becoming aware of the urgent need to prevent the exploitation of personal data, including medical records, [16]. In this work we consider a technical process for protecting medical information and other data assets using a technique of DNA cryptography further developed here. The personŠs data is protected using his own blood mineral levels as a seed for selecting, transmitting, and recovering that personŠs public key. As we know that the management of public keys remains a challenge, we will use a mechanism to generate the public key considering the specific individual blood analysis.

2 Hiding in DNA

According to Wikipedia ŞComputer security is a field of computer science concerned with the control of risks related to computer use.Ť Common use of the term Şcomputer securityŤ refers to several very important aspects of any computer-related system: confidentiality, integrity, availability, and authentication, [4]. Essential parts of what we may call data security, specifically confidentiality and authentication, are achieved using cryptography, [9], which has a long and fascinating history. The most complete non-technical account of the subject is David KahnŠs book, The Codebreakers. Public Key Cryptography is one set of cryptographic techniques for providing confidentiality, preventing data compromise, detecting alteration of data and verifying its authenticity, [11]. Recent research considers the use of the Human genome in cryptography. In 2000, the Junior Nobel Prize was awarded to a young Romanian-American student, Viviana Risca, for her work in DNA cryptography.

2.1 DNA Cryptography

A DNA encoded message is first camouflaged within the enormous complexity of human genomic DNA, and then further concealed by confining this sample to a microdot. A prototypical Śsecret messageŠ DNA strand contains an encoded message flanked by polymerase chain reaction (PCR) primer sequences (Fig. 1).To encode data characters in DNA, triplets were used in a simple substitution cipher (Fig. 2). Denatured human DNA provides a very complex background for concealing secret-message.

Figure 1: DNA coder, [13]

Viviana knowing both the secret-message DNA, PCR primer sequences and the encryption key could readily amplify the DNA and then she proposed a mechanism to read and decode the message, [13]. Use of the encryption key (Fig. 2) to decode the resultant DNA sequence (Fig. 3) yielded the encoded text, containing probably the most significant secret of the dawn of the microdot era: ŞJune 6 invasion: NormandyŤ, (Fig. 3).

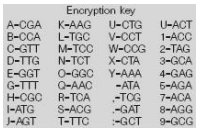


Figure 2: Key used to encode a message in DNA, [13]

Figure 3: June 6 invasion, [13]

We propose to encode the medical records of an individual in DNA data strand flanked by unique primer sequences, which we obtain in the process of deriving DNA Public Key from blood analysis. The specific minerals and the deviation from the normal values are considered as a first step. To improve the security mechanism primers from the synergetic minerals or other special primers are able to be used to change the original minerals primers. We then mix it among other decoy DNA strands that will together be sent to a receiver through a public channel.

2.2 Biological principles

DNA is a Genetic Program? This is a Rupert Sheldrake question, theoretical biologist that extends the Jung concepts from the collective human level to the biological, mineral, etc. levels, generally to the whole universe, [14]. August WeismannŠs theory of the germ-plasma (end of 19th century) is the ancestor of the present idea of genetic programming. The genetic program is assumed to be identical with DNA, the genetic chemical. The genetic information is coded in DNA and this code forms the genetic program. The central dogma of the molecular biology is illustrated in (Fig. 4). A DNA segment that constitutes a gene is read, starting from the promoter (starting position) of the DNA segment. The non-coding areas (intron) are removed according to certain tags remain coding areas (extron) are rejoined and caped. Then the sequence is transcribed into a single stranded sequence of mRNA (messenger RNA). The mRNA moves from the nucleus into the cytoplasm. In chromosomes, DNA acts as a template for the synthesis of RNA in a process called transcription. RNA Synthesis and Processing in the transcription and the splicing steps, introns are cut out, and extrons are kept to form mRNA, which will perform the translation work, [1].

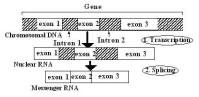


Figure 4: Central Dogma of Molecular Biology

In the translation process, codons are translated into the amino acids according to the genetic code. The introns are characterized by their starting and ending codes, which makes determining their location relatively easy. The DNA form of information is scanned by the Sender to find the locations of the introns; which he then records. He cuts out the introns according to the specified pattern, so that the DNA form of data is translated into its mRNA form, which then translates into protein form of data according to the genetic code table (61 codons to 20 amino acids).The protein form of data can be then transferred to the Recipient.The starting and pattern codes of the introns, their locations (or offsets), the removed introns, and the codon-amino acids mapping of the protein, are the keys to decrypt the protein form of data, and they can be transferred to the Recipient through a secure channel. When the protein form of data and the keys are received, the recipient identifies the secret data-carrying DNA strand using the program that associates the nucleotide sequence, with a specific mineral level based on the blood analysis of the individual. The Recipient obtains the unique primer sequences that mark the beginning and the end of the secret data DNA strand hidden among the decoy strands and using the information conversion program, reads the medical record of the individual.

3 Public Key Infrastructure (PKI) with DNA Cryptography

Introducing DNA cryptography into the common PKI scenario, it is possible to follow the pattern of PKI, while also exploiting the inherent massively-parallel computing properties of DNA bonding to perform the encryption and decryption of the public and private keys. The resulting encryption algorithm used in the transaction is much more complex than the one used by conventional encryption methods, [5]. To put this into terms of the common Stefani and Otto description of secure data transmission and reception, they are basing their argument of DNA cryptography on Otto providing Stefani his public key which will constitute each unique blood analysis and Stefani will use it to send an encryptedmessage to him (Fig. 5), [7]. A potential eavesdropper, Rya, will have a formidable amount of work to perform to attempt decryption of the transmission compared to either Stefani or Otto. Public key encryption splits the key up into a public key for encryption and a secret key for decryption, [6]. It's impractical to determine the secret key from the public key. Otto generates a pair of keys and tells everyone his public key, while only he knows his secret key.

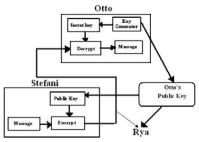


Figure 5: Public Key Encryption

Anyone can use Otto's public key to send him an encrypted message, but only Otto knows the secret key to decrypt it. This asymmetric scheme allows Stefani and Otto to communicate in secret without having to physically meet, unlike symmetric key encryption methods, [2]. A prototypical Śsecret messageŠ DNA strand contains an encoded message flanked by primer sequences. The intended recipient identifies the secret data-carrying DNA strand using the program that associates the nucleotide sequence with the specific mineral levels from the particular individualŠs blood analysis. He then obtains the unique primer sequences that mark the beginning and the end of secret data DNA strand hidden among the decoy strands. In this last step, he uses the information conversion program and reads the medical record of the individual. The DNA form of information, D, has length n, and is composed of k introns having average length m, thus, the mRNA form of information, DŠ has length: n-k*m. If Rya, an eavesdropper, can listen to Otto and StefaniŠs communication and tries brute force attack, it would be a very expensive computational problem for her. Since in the protein-to-mRNA step there are 61 coding codons and only 20 amino acids, in average there are 3 codons to be mapped onto the same amino acid. Such a brute force attack method could be impossible for Rya.

3.1 The Technique of Deriving DNA Public Keys from Blood Analysis

Etienne Guillé identified the nucleotides sequences that correspond to some minerals. As an example the Gold sequence is GA AT AG AC GC AA and is associated to the Sun as a correspondent planet [3]. The minerals are able to influence the DNA by conformational changes and also by activating or inhibit coding DNA sequences, Figure 6, (Gold ŰAu, Fe-Iron).

Mineral	Nucleotide Sequence
Fe	AT AG AC GG AA
Au	GA A T A G A C GC AA

Figure 6: Mineral-nucleotide correspondence table

As blood analysis results are specific for each person Figure 7, we can associate to a mineral such as Calcium, from the medical result, based on its concentration level a nucleotide sequence.This nucleotide sequence based on the medical results of the specific person will constitute the unique primer sequences. However, as we generate a sequence for calcium, an intruder knowing the calcium level could possibly discover the nucleotide sequence.

Mineral	Blood Analysis Result	Normal Level
Ca	2.81 mmol/l	2.25-2.7
Mg	0.89 mmo/l	0.75-1.05

Figure 7: Example blood mineral analysis table

Therefore, as an additional layer of security, we propose to associate each mineral with a corresponding mineral Ű which we will call here, a synergetic mineral pair Ű for example, Ca-Fe, Figure 8, or other special mineral established by the encryption process. Then, we will substitute the nucleotide sequence for calcium with that of iron. For our purposes, this step serves as an intermediate substitution table for increased obscurity. Thus a personŠs data-carrying DNA strand will be flanked by primer sequences unique to that individual. We will do the association in such way that from every most recent blood analysis result, a new primer sequence will be generated. After generating the unique primer sequence n-base primer will result.

Mineral	Paired synergetic minerals
Ca	Fe, Au, P
Mg	K, Ca, Na,

Figure 8: Paired mineral table

As we know that the management of public keys remains a challenge, we will use each unique blood analysis as the basis for a public key. The medical results will be of no use to an unauthorized person, and for an intruder, it would prove extremely difficult to read and detect the DNA strand that contains the medical history of a person, without knowing the specific unique primer sequences of the specific person. Performing a quick search by a program we will get the chosen mineral (M). Considering a dedicated medical application (for alternative therapy) developed by our research team, we consider the blood results of an individual. If the mineral level has the level (L) witch is not corresponding to a normal level (NL) and is equal to value: L=X.YZ mmol/l. Then: 1st step: We associate X times the nucleotide sequence corresponding to the synergetic mineral of the selected mineral getting a sequence S1. 2nd step: In the second step, to this sequence of nucleotide S1 we add Y+Z numbers of nucleotide from the original sequence of a synergetic mineral. Resulting sequence S will constitute the unique primer sequence. Else: If all blood results are in a normal level we will chose a minimal encoding to generate the unique primer sequences because this means we donŠt have anything to hide about an individual. This means we will just associate the nucleotide sequence corresponding to a chosen mineral.

4 Encryption algorithm using minerals

The main steps of the algorithm are: Step 1: Stefani (the sender) provides Otto (the receiver) her public key which will constitute each unique blood analysis of the specific person. Step 2: A secret DNA data strand contains 3 parts: - Secret DNA data strand in the middle - Unique primer sequences on each side Step 3: Stefani uses the technique of deriving DNA public key from blood analysis. 3.1 In this process Stefani uses a program which associates to a specific mineral the nucleotide sequence based on the medical results of a specific person which will constitute the unique primer sequences. Step 4: Using an information conversion program Stefani encodes the medical records of an individual in DNA data strand flanked by unique primer sequences and mixes it among other decoy DNA strands. Comments: Otto will use programs that perform the reverse processes as those performed by Stefani: simulating the transcription, splicing, and translation per the Central Dogma of Molecular Biology (CDMB) and performing a quick search by a program we will get the chosen mineral. 4.1 According to CDMB during the process of transcription Stefani cuts out the introns from the data-encoded DNA, resulting in Encryption key 1. E1= starting and pattern codes of introns, rezults C1=E1(P), where P is plaintext and C is the ciphertext. 4.2 Stefani translates the resulted spliced form of the data, results Encryption key 2, E2=the codon amino acids mapping, results C=E2(C1). 4.3 Stefani obtains the data-encoded protein after the translation process. 4.4 Stefani sends to Otto through a public channel the keys E1 and E2. Step 5: Stefani sends to Otto through a public channel the encoded protein form of the data. Step 6: Otto uses the key E2 to recover the mRNA form of the data from protein form of the data. Decryption key D1=E2, results P1=D1(C).Step 7: Otto recovers the DNA form of the data in the reverse order as Stefani encrypted it. Decryption key D2=E1, results P=D2(P1). Step 8: Otto identifies the secret data-carrying DNA strand using the program that associates the nucleotide sequence based on the blood to a specific mineral analysis of the particular individual. He obtains the unique primer sequences that mark the beginning and the end of secret data DNA strand hidden among the decoy strands. Step 9: In this last step, Otto uses the information conversion program and reads the medical record of the individual.

5 Conclusion

A lot of similarities between the spiritual domain and the technical domain are able to be realized. Some scientists are capable to understand and use the spiritual concepts, and the technical results will be more relevant, [19]. The complexity of the universe is like the complexity of human life. Each human being a unique, understanding in a profound mode that and the spiritual concepts, we will be able to develop relevant technical new solutions, including in biometrics cryptography. The software applications dedicated to different domains must respect nowadays a lot of security elements including a modularity facility. Considering a medical dedicated application (for alternative therapy) developed inside our research team, we try to adapt new security facilities using DNA cryptography. The DNA technology will be used to implement security facilities for software applications, [18]. The encryption keys available will be used considering all the possible combinations. We plan to do future work implementing such an alternate cryptography method. Analyzing in a deeper mode some spiritual concepts (eneagrams, tetractis, I Ching, mandalas, morphic theory, etc.) from oriental philosophy and other old civilizations, it is possible to consider special properties that are in conjunction with the cryptology domain. The trinity is the base in the human spiritual evolution and in the DNA structure. Some associations among metals, minerals, planets, biological and morphogenetic fields are able to offer new technical solutions in the future.

References

[1] Borem Aluízio Fabricio, R.Santos, 2003 “Understanding Biotechnology,” *Publisher: Prentice Hall*, PTRPub Date: January 17, 2003.

[2] Bajenescu T., Borda M. “Securitatea in informatica si telecomunicatii,” *Edit. Dacia*, 2001, Romanian language

[3] Bivolaru Gregorian “Enciclopedia naturista a elementelor minerale,” *Edit. Shambala*, 2003, Romanian language

[4] Bruce Schneier “Applied Cryptography,” *web published*

[5] Gehani Ashish, La Bean, Thomas H. Reif, JohnH, “DNA-Based Cryptography,” *Department of Computer Science, Duke University*, June 1999

[6] Cuvakin Anton, Cyrus Peikari, “Security Warrior,” *Publisher: O’Reilly*, Pub Date: January 2004

[7] Hodorogea Tatiana, Mircea-Florin Vaida “Alternate Cryptography Techniques,” *ICCC 2005*, 2005

[8] Howard Michael, David LeBlanc “Writing Secure Code,” *Publisher: Microsoft PressPub*, December 4, 2002.

[9] Garfinkel Simson “Web Security, Privacy and Commerce, second Edition,” *Publisher: OŠReilly*, November 2001

[10] Garfinkel Simson, Lorrie Faith Cranor, “Security and Usability” *Publisher: OŠReilly*, August 2005

[11] Kahn D., “The Codebrakers,” *McMillan, New York*, 1967

[12] Kang Ning “A Pseudo DNA Cryptography Method Independent Research Study Project for CS5231 ,” *Method Independent Research Study Project for CS5231*

[13] Taylor Clelland Catherine, Viviana Risca,Carter Bancroft, 1999 Nature Magazine Vol.. 399, “Hiding Messages in DNA Micodots,” *Nature Magazine Vol.. 399*, June 10, 1999.

[14] Shaldrake Rupert, “Mind, Memory, and Archetype: Morphic Resonance and the Collective Unconscious,” *web published*

[15] Striletchi Cosmin, Mircea-Florin Vaida, “Security Techniques for Enhancing Web Distributed Applications,” *Second IASTED International Conference on Communications,Internet, and Information Technology, CIIT2003,* , web published.

[16] Vaida Mircea-Florin and colab., “ Java 2 Enterprise Edition (J2EE). Aplicatii multimedia,” *Editura Albastra,Cluj-Napoca*, 2002.

[17] Vaida Mircea-Florin “Information Society Development and Human Evolution,” *Baile Felix, ICCC 2004, pp. 414-420*, May 27-29, 2004,

[18] Vaida Mircea-Florin “Security and Java,” *Conference at the Université de Savoie, France, supported by Shuffle project*, May 2004

[19] Vaida Mircea-Florin "Teaching Computers As a Human Spiritual Evolution," *the 4th IASTED International Conference on Web-Based Education, Grindelwald, Switzerland, pp. 667-672*, 2005.

Tatiana Hodorogea, Mircea-Florin Vaida
Technical University of Cluj-Napoca,
Faculty of Electronics and Telecommunications
Address: Baritiu Street No.26, 3400, Cluj-Napoca, Romania
E-mail: thodorogea@yahoo.com, Mircea.Vaida@com.utcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 268-273

Formalizing Peer-to-Peer Systems based on Content Addressable Network

Adrian Iftene, Gabriel Ciobanu

Abstract: Peer-to-peer systems are preferred to the centralized systems because their architecture facilitates fault tolerance, availability, scalability and performance. In this paper we present a formal description of the peer-to-peer system implemented with Content Addressable Network architecture. We describe this system using a variant of the distributed π-calculus called *P2P π-calculus*. We define a new bisimulation for *well-located systems* called *go-bisimulation* and give few related results.

Keywords: Peer-to-peer system, Content Addressable Network, process algebra, bisimulation.

1 Introduction

The popularity of distributed file systems continues to grow significantly in the last years, due to the success of peer-to-peer services like Napster, Gnutella, Kazaa and Morpheus. Peer-to-peer systems offer a decentralized, self-sustained, scalable, fault tolerant and symmetric network of machines providing an effective balancing of storage and bandwidth resources. In this paper we refer to a peer-to-peer system *completely distributed* (it requires no form of centralized control, coordination or configuration), *scalable* (its nodes control only a certain region which is a part of the whole system and does not depend on the total number of nodes in the system), and *fault tolerant* (it can avoid some failures).

P2P Systems:

The term Peer-to-Peer (P2P) refers to a class of systems (hardware and software) which employ distributed resources to perform a function in a decentralized manner. Each node of such a system has the same responsibility. Basic goals are decentralization, immediate connectivity, reduced cost of ownership and anonymity. P2P systems are defined in [1] as "*applications that take advantages of resources (storage, cycles, content, human presence) available at the edges of the Internet*". The P2P architecture can help to reduce storage system costs and allow cost sharing by using the existing infrastructure and resources of several sites. Considering these factors, the P2P systems could be very useful in designing the future generation of distributed file systems.

CAN:

The Content Addressable Network (CAN) provides a suitable architecture for completely distributed, scalable, and fault-tolerant system [7]. We think that CAN can enable new communication models and applications. CAN for is used mainly like a distributed hash table. We consider CAN as part of a virtual d-dimensional Cartesian coordinate space. This is a logical space, and it is not related to a specific physical system. In this d-dimensional coordinate space, two nodes are neighbours if their coordinates coincide along $(d-1)$ dimensions and differ in only one dimension. The basic operations performed on a CAN are insertion and deletion of nodes. In this paper we refer to the procedure of inserting a new node. A new node joining an existing CAN should receive its own portion of the space. **CAN Insertion** starts when a new node looks for a *source node* which is already in CAN. After that, the new node learns the number of CAN dimensions and generates a d-point in this space (this point is called the *destination node*). A route from the *source node* to the *destination node* follows the straight path through the Cartesian space (see the figure below). The *destination node* splits its zone in half, and assigns one half to the new joining node. Finally, the neighbours of the split zone must be notified about the new node. When a node disappears from CAN, we should ensure that its region is transferred to the remaining neighbours. In CAN Deletion, if there is only a neighbour, it takes control of the remaining region. If there are more than one neighbour, then the remaining region is split in smaller zones which are distributed to the neighbours according to a certain procedure.

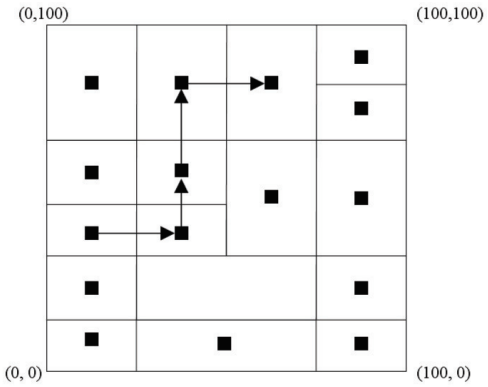


Figure 1: CAN movement

2 Problem

A CAN region HP can be uniquely identified either by an associated d-dimensional hyper-parallelepiped and by its direct neighbours $HP^1, HP^2, ..., HP^{nv}$. A hyper-parallelepiped $HP \subseteq S \subset R^d$ has the shape given by $[min_1, max_1] \times [min_2, max_2] \times \ldots [min_d, max_d]$, where S denotes the initial region associated at the whole network, and nv represent the number of direct neighbours of the current region HP. In this d-dimensional space a hyper-point has the form $p = (p_1, p_2, ..., p_d)$. From now on we use the word region to denote the associated hyper-parallelepiped, and the word point to denote the hyper-point given by the center of a region. Whenever we want to test if a point p is in region HP, we should test d times if $p_i \in [min_i, max_i]$, for $1 \leqslant i \leqslant d$. In this space S the movement from a region HP^c (where is the source node) to a point p^{dn} (where is the destination node) requires a calculation of a *minimal distance* from a point to a region. This distance is calculated by using the distance from the point to the center of the corresponding region:

HP^c center is given by $cHP^c = (\frac{max_1^c - min_1^c}{2}, \frac{max_2^c - min_2^c}{2}, \ldots, \frac{max_d^c - min_d^c}{2})$,

and the distance is $d(p^{dn}, cHP^c) = \sqrt{(p_1^{dn} - cHP_1^c)^2 + (p_2^{dn} - cHP_2^c)^2 + \cdots + (p_d^{dn} - cHP_d^c)^2}$

Let us consider that at a certain moment we are in a region HP^c (where is our source node p^s), and we want to move to a region HP^{dn} (where is our destination node). We take all the neighbours $HP^1, HP^2, ..., HP^{nv}$ of HP^c, and we calculate $d(p^{dn}, cHP^i)$ for each of them. We move from the current region HP^c to the HP^{min} which satisfies: $d(p^{dn}, cHP^{min}) = min\{d(p^{dn}, cHP^i) \mid 1 \leqslant i \leqslant nv\}$. If we have many regions satisfying this equality, then we use the lexicographic order to select the region with greatest coordinates over the corners $\{(max_{1i}, max_{i2}, \ldots, max_{id}) \mid 1 \leqslant i \leqslant nv\}$ of the regions.

This is the ideal situation when we have not dead or overloaded regions on path from the source node to the destination node. These failure regions do not respond at the identification messages, and they should be by-passed on the way to the destination (see the figure below). This is the reason why we should relax the condition to select always the minimal path. This new requirement is implemented by using a tabu-list which is kept in the most recent visited nodes. A new node is selected among the points of the neighbours, excepting the nodes in the tabu-list of the current node. The resulting paths achieved in this way are not always optimal, but these paths allow to avoid the failure regions of the network. It is also possible that we cannot detect a path from the source node to the destination node. This fact is expressed in our algorithm through a stop condition when after a specified number of steps we do not reach or come closer to the destination node.

Here we present the ideal case, and we specify some additional remarks to suggest what should be done to avoid the failure region.

3 Syntax and semantic of P2P π-calculus

Distributed π-calculus (Dπ) is an extension of the π-calculus with explicit notions for localization and code migration [3]. In Dπ communication is local, and messages to the remote resources have a clear itinerary. We adapt distributed π-calculus [3], and modify it by including elements specific to P2P; the resulting calculus is called P2P

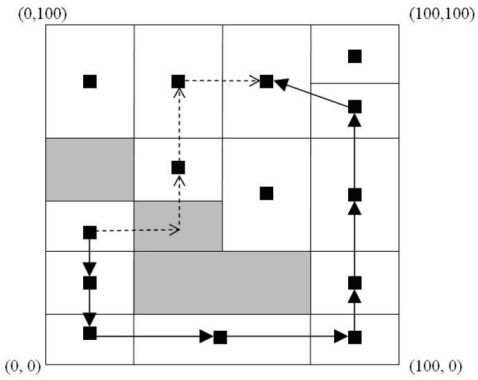


Figure 2: CAN movement when we have failure regions

π-calculus. Some elements of this new calculus are taken from a process algebra used in PEPITO[1] project [2]; this part is used to describe the static aspects of the network. We use additional elements from the distributed π-calculus to describe the dynamic part of network.

Syntax

Values and Expressions: we use a set $\mathcal{V}$ of values: integers, lists, and "undefined value" $\perp$. We define an evaluation over the set $\mathcal{E}$ of expressions by $[.]:\mathcal{E} \rightarrow \mathcal{V}$.
Names: we assume a countable set $\mathcal{N}$ of simple names which is divided in set $N_{ch} = \{a, b, \ldots\}$ of *channel names* and set $N_{loc} = \{l, k, HP, \ldots\}$ of *location names*. We assume that *composed names* have the form $a@l$ meaning a channel a at location l. We use $n, m, w, \ldots$ to denote simple and composed names. Operation $\_@l$ over simple and composed names is defined by:

$$n@l = \begin{cases} a@l & \text{if } n = a \\ n & \text{if } n = a@l' \end{cases}$$

Types: We have the following types of the names:

$$\begin{array}{ll} (\ \textit{general types}\ ) & \sigma ::= \gamma \mid v \mid \gamma@\lambda \\ (\ \textit{channel types}\ ) & \gamma ::= Ch(\sigma_1, \sigma_2, \ldots) \\ (\ \textit{location types}\ ) & \lambda ::= \{a_1 : \gamma_1, \ldots, a_n : \gamma_n\} \end{array}$$

Location types $\lambda ::= \{a_1 : \gamma_1, \ldots, a_n : \gamma_n\}$ are introduced in [3]. Intuitively $a_1, ..., a_n$ correspond to the available channel resources, and $\gamma_1, \ldots, \gamma_n$ correspond to the available locations resources.

Values $\mathcal{V}$	$v ::= 0, 1, 2, \ldots \mid [\,] \mid \perp \mid v_1 :: v_2$
Expressions $\mathcal{E}$	$e ::= v \mid x \mid head(e) \mid tail(e)$
Boolean tests B	$\phi ::= e_1 \leq e_2 \mid p \in HP$
Input types G	$G ::= 0 \mid a(\tilde{x}).P \mid G + G$
Processes $\mathcal{P}$	$P, Q ::= G$
$(asynchronous \quad output)$	$\mid \overline{a}\langle\tilde{x}\rangle$
$(parallelism \quad and \quad restriction)$	$\mid P \mid P \mid (P)\backslash a$
$(definition \quad and \quad movement)$	$\mid A\langle\tilde{e}\rangle \mid go \quad l.P$
$(process \quad names \quad generation)$	$\mid (\nu w)P$
$(assignment and conditional)$	$\mid x := e \mid if \quad \phi \quad then \quad P \quad else \quad Q$
Systems $\mathcal{S}$	$S ::= [l :: P] \mid (S \mid S') \mid (\nu w)S$

Table 1: Syntax of P2P π-calculus

Static and Operational Semantics

Expressions evaluation is given by:

[1] http://www.sics.se/pepito

$$[e] := \begin{cases} v & \text{if } e = v \in \mathcal{V} \\ v & \text{if } e = x \text{ and } eval(x) = v \\ v_1 & \text{if } e = head(e') \wedge [e'] = v_1 :: v_2 \\ v_2 & \text{if } e = tail(e') \wedge [e'] = v_1 :: v_2 \\ \perp & \text{otherwise} \end{cases}$$

Boolean evaluation is given by:

$[e_1 \leq e_2]$ is *true* iff $\perp \neq [e_1] \leq [e_2]$

$[p \in HP^c]$ is *true* iff: $p_i \in [min_i^c, \max_i^c]$, for all $i = \overline{1,d}$

Substitution of variables $\tilde{x}$ by values $\tilde{v}$ in a process P is written $P[v_1/x_1,...,v_n/x_n]$, and is done for all unbound appearance of $\tilde{x}$ in P. The variables $\tilde{v}$ are bound in $a(\tilde{v}).P$.

A reduction relation $>$ is a relation over processes given by the following rules:

1. $\overline{a}\langle\tilde{e}\rangle > \overline{a}\langle\tilde{v}\rangle$ if $[\tilde{e}] = \tilde{v}$.
2. $A\langle\tilde{e}\rangle > P[v_1/x_1,...,v_n/x_n]$ if $A(\tilde{e}) = P, |\tilde{e}| = |\tilde{x}| = n, [\tilde{e}] = \tilde{v}$.
3. if ϕ then P else $Q > P$ if $e_b(\phi)$.
4. if ϕ then P else $Q > Q$ if $\neg e_b(\phi)$.
5. $l' :: go\ l.P > l :: P$.

A set of actions $\mathcal{A}$ is defined by $\mu ::= \tau \,|\, a\tilde{v} \,|\, \overline{a}\tilde{v}$ with the following semantic meaning: τ represents an internal action, $a\tilde{v}$ represent waiting for values tuple $\tilde{v}$, and $\overline{a}\tilde{v}$ represents sending a value tuple $\tilde{v}$. A channel ch is associated for each action; $ch : \mathcal{A} \to \mathcal{N} \bigcup \{\perp\}$ is defined by $ch(\tau) : = \perp$, $ch(a\tilde{v}) : = a$ and $ch(\overline{a}\tilde{v}) : = a$. Operational semantic of P2P π-calculus is given by a transition relation $\to \subseteq \mathcal{S} \times A \times \mathcal{S}$ defined by the following rules:

$$(IN)\frac{}{a(\tilde{x}).P \xrightarrow{a\tilde{v}} P[v_1/x_1,...,v_n/x_n]} \text{ if } |\tilde{v}| = |\tilde{x}| \quad (OUT)\frac{}{\overline{a}\langle\tilde{v}\rangle \xrightarrow{\overline{a}\tilde{v}} 0} \quad (RES)\frac{P \xrightarrow{\mu} P'}{(P)\backslash a \xrightarrow{\mu} (P')\backslash a} \text{ if } a \neq ch(\mu)$$

$$(COM_L)\frac{P \xrightarrow{a\tilde{v}} P', Q \xrightarrow{\overline{a}\tilde{v}} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \quad (PAR_L)\frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \quad (SUM_L)\frac{G_1 \xrightarrow{a\tilde{v}} P'}{G_1 + G_2 \xrightarrow{a\tilde{v}} P'} \quad (RED)\frac{P > Q, Q \xrightarrow{\mu} Q'}{P \xrightarrow{\mu} Q'}$$

Structural congruence $\equiv$ is a standard notion of equivalence used to identify processes and systems according to their syntactic structure [6].

4 Go-bisimulation

We use three different types of name generation: channel name generation (νa), location name generation $(\nu a@l)$ and location name generation with a specific type $(\nu l : \lambda)$. We denote by $(\nu w)P$ a local name w for a located process P, and by $(\nu w)S$ a local name w for a distributed system S.

$l' :: go\ l.P$ represents a movement of process P from location l' to location l. Such a movement is used in a P2P system with a CAN architecture when we should insert a new node. We start from an initial region HP^s (where we have the source node p^s), and we generate randomly one point p^{dn} (destination node). For every current node HP^c (initial we have $HP^c = HP^s$) we build the process expressions which describe the following steps:

1. $P_0(HP^c) :\ min := c$
2. $P_i(HP^c) :\ HP^c :: go\ HP^i$. if $(d(p^{dn}, HP^i) \leq d(p^{dn}, HP^{min}))$ then $min := i$. Q else 0,
   for $1 \leqslant i \leqslant nv$, where nv is the number of neighbours of current node
3. Q : if $(p^{dn} \in HP^{min})$ then 0 else $P_0(HP^{min})$

The first process P_0 identifies the current region on the optimal path to the destination node. The processes P_i defines the visits to all the neighbours in order to find the minim distance to the destination node and to update the new region of the optimal path to the destination node. The last process invokes a call of process P_0 with the new detected region as a current region on the optimal path to the destination node. The iteration ends when the region of the destination node is detected.

When we insert a new node, the dynamics of a P2P system is mainly based on the movements from one region to another in order to find an optimal path to the region of the new node. We denote by $HP \xrightarrow{go,p} HP'$ a movement from a region HP to a region HP' in order to find an optimal path to point p according to the procedure described above. Moreover, we denote by $HP \xrightarrow{go,p}_* HP'$ a sequence of movements from HP to HP' in order to find an optimal path to point p if there is $n > 0$ such $HP = HP^0 \xrightarrow{go,p} \cdots \xrightarrow{go,p} HP^n = HP'$.

Proposition 1 (Location confluence). If $HP^c \xrightarrow{go,p}_* HP^1$ and $HP^c \xrightarrow{go,p}_* HP^2$, and there is an optimal path to the destination node p belonging to a region HP^d, then we have $HP^1 \xrightarrow{go,p}_* HP^d$ and $HP^2 \xrightarrow{go,p}_* HP^d$.

Proof. According to the condition imposed by step 2 of the procedure, a movement from one region to another along an optimal path means a decreasing of the distance to point p. Hence we have $d(p,HP^c) > d(p,HP^1)$ and $d(p,HP^c) > d(p,HP^2)$. We have $d(p,HP) \geq 0$ for every region HP, and the distances from regions HP^1 and HP^2 to the point p describe a descendant sequence to value 0. HP^1 and HP^2 are on different trajectories leading to the region HP^d where point p is located. Therefore both trajectories should end in the same region HP^d.

It is possible to have problems in finding an optimal way to the destination node whenever we have some failure regions. This is mainly because we cannot find a region having a smaller distance to the destination node than the distance from the current region. Then we select a region HP^{new} with a minimal $d(p,HP^{new})$ greater than the current distance $d(p,HP^c)$. We add HP^{new} to the tabu-list of the current node, and we also add HP^c to the tabu-list of the new node. In this way we eliminate the possible cycles from HP^{new} to HP^c. It is not sure that we can find a path to the destination node. In order to avoid a never-ending process of search and movement, we impose a stop condition after a number of steps without finding a decreasing distance to the destination node.

Definition 1 (go-LTS) A *go-Labelled Transitional System go-LTS* is a pair (Q,T) composed from a set of regions $Q = \{HP^0, HP^1, ...\}$ and a transition relation $T \subseteq (Q \times Points \times Q)$. $(HP,p,HP') \in T$ is denoted by $HP \xrightarrow{go,p} HP'$.

Definition 2 (go-Bisimulation)

a. *Let (Q,T) be a go-LTS and $\mathcal{S} \subseteq (Q \times Q)$. Then $\mathcal{S}$ is a* **go-simulation** *over (Q,T) related to p if for every $HP\mathcal{S}HP'$: if $HP \xrightarrow{go,p} HP^1$ then it exists HP'^1 such that we have $HP' \xrightarrow{go,p} HP'^1$ and $HP'\mathcal{S}HP'^1$. We say that HP' is a go-simulation for HP related to p.*

b. *Let $\mathcal{B}$ be a binary relation over Q. $\mathcal{B}$ is a* **go-bisimulation** *related to p if both $\mathcal{B}$ and $\mathcal{B}^{-1}$ are go-simulations related to p. We say such HP and HP' are go-bisimilar related to p, and denote $HP \sim_p HP'$, if there exist a go-bisimulation related to p such that we have $HP\mathcal{B}HP'$.*

The go-simulation binary relation links the paths from two regions to a destination point. If these paths have the same number of steps, then we have a go-bisimulation.

Proposition 2.

a. $\sim_p$ is an equivalence relation.

b. $\sim_p$ is itself a go-bisimulation related to p.

Proof. (1) An equivalence relation requires reflexivity, symmetry and transitivity.
For reflexivity, it is enough to prove that identity relation over Q, namely $Id_Q = \{(HP,HP) \mid HP \in Q\}$ is a go-bisimulation related to p. Because $Id_Q = Id_Q^{-1}$ it is enough to prove that Id_Q is a go-simulation related to p. This means that for every HP with $HPId_QHP$, if $HP \xrightarrow{go,p} HP^1$ then exist HP'^1 such that $HP \xrightarrow{go,p} HP'^1$ and $HP^1 Id_Q HP'^1$. Of course $HP'^1 = HP^1$ satisfy both conditions.
For symmetry, we must prove that if $\mathcal{B}$ is a go-bisimulation, then $\mathcal{B}^{-1}$ is also a go-bisimulation. Since $\mathcal{B}$ is a go-bisimulation, then we have $\mathcal{B}$ and $\mathcal{B}^{-1}$ are go-simulations. This is equivalent with the fact that $\mathcal{B}^{-1}$ and $(\mathcal{B}^{-1})^{-1}$ are go-simulations. Therefore $\mathcal{B}^{-1}$ is a go-bisimulation.
For transitivity, we must prove that if $\mathcal{B}_1$ and $\mathcal{B}_2$ are go-bisimulations, then their composition $\mathcal{B}_1\mathcal{B}_2 = \{(HP,HP") | \exists HP'$ such that $HP\mathcal{B}_1HP'$ and $HP'\mathcal{B}_2HP"\}$ is also a go-bisimulation. It is enough to show that this is a go-simulation. Let $(HP,HP") \in \mathcal{B}_1\mathcal{B}_2$, and $HP \xrightarrow{go,p} HP^\sharp$. Since there exists HP' such that $HP\mathcal{B}_1HP'$ and $HP'\mathcal{B}_2HP"$, then there exists $HP'^\sharp$ such that $HP' \xrightarrow{go,p} HP'^\sharp$ and $HP^\sharp\mathcal{B}_1HP'^\sharp$, and also $HP"^\sharp$ such that $HP" \xrightarrow{go,p} HP"^\sharp$ and $HP'^\sharp\mathcal{B}_2HP"^\sharp$. Thus $(HP^\sharp, HP"^\sharp) \in \mathcal{B}_1\mathcal{B}_2$, which means that $\mathcal{B}_1\mathcal{B}_2$ is a go-simulation.
(2) Let be $HP \sim_p HP'$. Then from definition we have $HP\mathcal{B}HP'$ for a go-bisimulation $\mathcal{B}$ related to p. Accordingly, if $HP \xrightarrow{go,p} HP^1$, then it exists HP'^1 for which we have $HP' \xrightarrow{go,p} HP'^1$, and $HP^1\mathcal{B}HP'^1$. From this we infer that $HP^1 \sim_p HP'^1$. Thus $\sim_p$ is a go-simulation, and also its inverse is a go-simulation.

The failure regions can evolve in time. This means that we can speak about static and dynamic failure regions. Static failure regions which can be by-passed preserve $\sim_p$ as an equivalence relation. The dynamic failure regions require a re-verification of the go-bisimulation. One situation is given by the nodes which previously satisfy the go-bisimulation, but now are down and induce failure regions. Therefore we should find other regions able to preserve the go-bisimulation relation. Another situation is when a failure region becomes active. The go-bisimulation relation can remain the same.It is also possible that the former paths are not optimal anymore, and we could define new go-bisimulation relations.

5 Conclusions and Future Work

In order to conclude, we can say that in this article we present a location confluence of a P2P specific architecture called CAN with respect to the insertion of a new P2P node, and we define go-bisimulation using a formalism called P2P π-calculus. We formalize the movement in an ideal P2P system. Then we describe what should be done to avoid failure regions. This presentation is a first step in formalizing P2P systems based on CAN architecture. We intend to extend it by adding timers in order to catch the fail situations when a node does not respond for a certain period of time.

References

[1] S. Androutsellis-Theotokis, and S. Spinellis, "A Survey of Peer-to-Peer File Sharing Technologies," *Athens University of Economics and Business White Paper* (WHP-02-03), 2002.

[2] J. Borgstrom, and U. Newstmann, "Verifying a Structured Peer-to-Peer Overlay Network: The Static Case," *Technical Report*, EU Project PEPITO, IC, 2004.

[3] M. Hennessy, and J. Riely, "Information Flow vs. Resource Access in the Asynchronous π-calculus," *ACM Transactions on Programming Languages and Systems*, Vol.24, No.5, pp. 566-591, 2002.

[4] A. Ingolfsdottir, "Semantic Models for Communicating Processes with Value Passing," *University of Sussex, Technical Report 8/94*, 1994.

[5] R. Milner, "*Communicating and Mobile Systems: the π-calculus*", Cambridge University Press, 1999.

[6] R. Milner, J. Parrow, and D. Walker, "A Calculus of Mobile Processes, Part.I/II," *Information and Computation*, 100, pp. 1-77, 1992.

[7] S. Ratnasamy et. al., "A Scalable Content Addressable Network", *Proceedings SIGCOMM*, ACM Press, pp.161-172, 2001.

Adrian Iftene, Gabriel Ciobanu
"Al. I. Cuza" University, Faculty of Computer Science
Address: General Berthelot, 16, 700483, Iasi
E-mail: adiftene@infoiasi.ro, gabriel@iit.tuiasi.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 274-279

General Information Dispersal Based on the Chinese Remainder Theorem

Sorin Iftene

Abstract: *Information dispersal* was introduced by Rabin [18] as a solution for the problem of splitting an information into n fragments such that the information can be reconstructed from any k fragments ($1 \leq k \leq n$). Béguin and Cresti [2, 3] extended it to more general access structures. In this paper we present a new general information dispersal scheme based on the Chinese remainder theorem.

Keywords: information dispersal, secret sharing, Chinese remainder theorem

1 Introduction and Preliminaries

Information dispersal deals with splitting an information into fragments such that the information can be reconstructed from some predetermined sets of fragments. Information dispersal was introduced by Rabin [18] for the *threshold* case. In this case, only the number of fragments is important for reconstructing the information. Béguin and Cresti [2, 3] extended it to more general access structures.

Besides the straightforward applications in the field of information storage and retrieval, information dispersal can be used, as proposed by Krawczyk [13] and later by Béguin and Cresti [2, 3], for designing computational secure secret sharing schemes.

The paper is organized as follows. The rest of this section is dedicated to some preliminaries on number theory, focusing on the Chinese remainder theorem. In Section 2, after an introduction to information dispersal, we present a new general information dispersal scheme based on the Chinese remainder theorem and we indicate how to select the parameters of the scheme in order to obtain short fragments. In Section 3 we present the application of general information dispersal in designing computational secure secret sharing scheme. The last section concludes the paper.

In the rest of this section we present some basic facts on number theory. For more details, the reader is referred to [6].

Let $a, b \in \mathbf{Z}$, $b \neq 0$. The *quotient* of integer division of a by b will be denoted by a `div` b and the *remainder* will be denoted by a `mod` b.

Let $a_1, \ldots, a_n \in \mathbf{Z}$ such that $a_1^2 + \cdots + a_n^2 \neq 0$. The *greatest common divisor* (gcd) of $a_1, \ldots, a_n$ will be denoted by $(a_1, \ldots, a_n)$.

Let $a_1, \ldots, a_n \in \mathbf{Z}$ such that $a_1 \cdots a_n \neq 0$. The *least common multiple* (lcm) of $a_1, \ldots, a_n$ will be denoted by $[a_1, \ldots, a_n]$.

Let $a, b, m \in \mathbf{Z}$. We say that a and b are *congruent* modulo m, and we use the notation $a \equiv b \ mod \ m$, if $m|(a-b)$. $\mathbf{Z}_m$ denotes the set $\{0, 1, \ldots, m-1\}$.

The Chinese remainder theorem has many applications in computer science (see, for example, [9]). We only mention its applications to the *RSA* decryption algorithm as proposed by Quisquater and Couvreur [17], the discrete logarithm algorithm as proposed by Pohlig and Hellman [16], and the algorithm for recovering the secret in the Mignotte's threshold secret sharing scheme [14] or in the Asmuth-Bloom threshold secret sharing scheme [1]. Several versions of the Chinese remainder theorem have been proposed. The next one is called the *general* Chinese remainder theorem [15]:

Theorem 1. *Let $k \geq 2$, $m_1, \ldots, m_k \geq 2$, and $b_1, \ldots, b_k \in \mathbf{Z}$. The system of equations*

$$\begin{cases} x & \equiv & b_1 \ mod \ m_1 \\ & \vdots & \\ x & \equiv & b_k \ mod \ m_k \end{cases}$$

has solutions in $\mathbf{Z}$ if and only if $b_i \equiv b_j \ mod \ (m_i, m_j)$ for all $1 \leq i, j \leq k$. Moreover, if the above system of equations has solutions in $\mathbf{Z}$, then it has an unique solution in $\mathbf{Z}_{[m_1, \ldots, m_k]}$.

When $(m_i, m_j) = 1$, for all $1 \leq i < j \leq k$, one gets the *standard* version of the Chinese remainder theorem. Garner [11] found an efficient algorithm for this case and Fraenkel [10] extended it to the general case.

2 General Information Dispersal Based on the Chinese Remainder Theorem

We present first some basic facts about information dispersal schemes. Béguin and Cresti [2, 3] have introduced the general information dispersal schemes.

Definition 2. *Let n be an integer, $n \geq 2$ and*[1] $\mathcal{A} \subseteq \mathcal{P}(\{1,2,\ldots,n\})$. *An* $\mathcal{A}$-information dispersal scheme *is a method of generating $(S,(F_1,\ldots,F_n))$ such that for any $A \in \mathcal{A}$, the problem of finding the element S, given the set of elements $\{F_i | i \in A\}$, is "easy".*

$\mathcal{A}$ will be referred to as the *access structure*, S will be referred to as the *information* and $F_1,\ldots,F_n$ will be referred to as the *fragments* of S.

A natural condition is that an access structure $\mathcal{A}$ is *monotone*, i.e.,

$$(\forall B \in \mathcal{P}(\{1,2,\ldots,n\}))((\exists A \in \mathcal{A})(A \subseteq B) \Rightarrow B \in \mathcal{A}).$$

Any monotone access structure $\mathcal{A}$ is well specified by the set of the minimal sets, i.e., the set

$$\mathcal{A}_{min} = \{A \in \mathcal{A} | (\forall B \in \mathcal{A} \setminus \{A\})(\neg B \subseteq A)\}.$$

An important particular case is that of the *threshold* information dispersal schemes. In these schemes, only the cardinality of the sets of fragments is important for recovering the information. More exactly, if the required threshold is k, $1 \leq k \leq n$, the minimal access structure is

$$\mathcal{A}_{min} = \{A \in \mathcal{P}(\{1,2,\ldots,n\}) \mid \ |A| = k\}.$$

In this case, an $\mathcal{A}$-information dispersal scheme will be referred to as an (k,n)-*information dispersal scheme*. Threshold information dispersal schemes were introduced by Rabin [18]. We present next a simpler version of Rabin's threshold information dispersal scheme, as described in [3].

- The information S is chosen as the vector of the coeficients of a random polynomial $P(x)$ of degree $k-1$ over the field of the positive integers modulo a prime q, $q \geq n$;

- The fragments $F_1,\ldots,F_n$ are chosen as $F_i = P(i)$, for all $1 \leq i \leq n$;

- Having the fragments $\{F_i | i \in A\}$, for some set A with $|A| = k$, the polynomial $P(x)$ can be obtained using Lagrange's interpolation formula as

$$\sum_{i \in A}(F_i \cdot \prod_{j \in A \setminus \{i\}} \frac{x-j}{i-j}).$$

Béguin and Cresti proposed the following general information dispersal algorithm [2, 3]:

- Choose, for $1 \leq i \leq n$, some positive integers p_i and q_i such that

$$(\forall A \in \mathcal{A}_{min})(\sum_{i \in A} \frac{p_i}{q_i} \geq 1) \tag{1}$$

and let $k' = [q_1,\ldots,q_n]$ and $n' = k' \cdot \sum_{i=1}^{n} \frac{p_i}{q_i}$ (remark that $k' \leq n'$);

- Use an (k',n')-threshold information dispersal scheme for constructing the fragments $F'_1,\ldots,F'_{n'}$ corresponding to the information S;

- Define the fragments $F_1,\ldots,F_n$ as $F_i = \{F'_j | j \in P_i\}$, where $\{P_1,\ldots,P_n\}$ is an arbitrary partition of the set $\{1,2,\ldots,n'\}$ such that $|P_i| = k' \cdot \frac{p_i}{q_i}$, for all $1 \leq i \leq n$;

[1] $\mathcal{P}(\{1,2,\ldots,n\})$ is the set of all subsets of the set $\{1,2,\ldots,n\}$.

- Having the fragments $\{F_i | i \in A\}$, for some minimal set A, the information S can be recovered using the reconstruction algorithm from the (k', n')-threshold information dispersal scheme. Indeed, by the choice of the fragments F_i and of the elements p_i and q_i, we will successively obtain that

$$\begin{aligned} |\cup_{i\in A} F_i| &= |\cup_{i\in A} \{F'_j | j \in P_i\}| \\ &= \sum_{i\in A} |P_i| \\ &= k' \cdot \sum_{i\in A} \frac{p_i}{q_i} \\ &\geq k' \end{aligned}$$

and, thus, at least k' elements from the set $\{F'_1, \ldots, F'_{n'}\}$ can be gathered.

A possibility of choosing the elements p_i and q_i such that relation (1) holds is $p_i = 1$ and $q_i = \min(\{|A| \mid A \in \mathcal{A}_{min} \wedge i \in A\})$. Indeed, in this case, for a minimal set A, we will obtain that $q_i \leq |A|$, for all $i \in A$ and thus $\sum_{i\in A} \frac{p_i}{q_i} \geq \sum_{i\in A} \frac{1}{|A|} = 1$.

We propose the following general information dispersal scheme based on the Chinese remainder theorem.

- The information S is chosen as an arbitrary positive integer such that $S < min_{A\in\mathcal{A}_{min}}([\{m_i | i \in A\}])$ where $m_1, \ldots, m_n$ are arbitrary positive integers;
- The fragments $F_1, \ldots, F_n$ are chosen as $F_i = S \bmod m_i$, for all $1 \leq i \leq n$;
- Having a set of fragments $\{F_i \mid i \in A\}$ for some $A \in \mathcal{A}$, the information S can be obtained as the unique solution modulo $[\{m_i | i \in A\}]$ of the system of equations

$$\left\{ \; x \equiv F_i \; mod \; m_i, \quad i \in A. \right.$$

Indeed, the information S is the unique solution modulo $[\{m_i | i \in A\}]$ of the above system of equations because S is an integer solution of the system by the choice of the fragments $F_1, \ldots, F_n$ and, moreover, $S \in \mathbf{Z}_{[\{m_i | i\in A\}]}$, because $S < min_{A\in\mathcal{A}}([\{m_i | i \in A\}])$.

The next example illustrates this scheme.

Example 3. Let $n = 4$ and $\mathcal{A}_{min} = \{\{1,2\},\{3,4\}\}$. Let consider $m_1 = 9$, $m_2 = 16$, $m_3 = 12$, and $m_4 = 18$. Suppose that the information is $S = 61$. We obtain the fragments $F_1 = 7$, $F_2 = 13$, $F_3 = 1$, and $F_4 = 7$. If we have the first two fragments, the information S can be obtained as the unique solution modulo 144 of the system of equations

$$\left\{ \begin{array}{lcl} x & \equiv & 7 \; mod \; 9 \\ x & \equiv & 13 \; mod \; 16 \end{array} \right.$$

An important issue in information dispersal is the size of fragments with respect to the size of the information. In our scheme, by finding $m_1, \ldots, m_n$, and the biggest $\alpha \geq 1$ such that

$$(max(m_1, \ldots, m_n))^\alpha < S < min_{A\in\mathcal{A}_{min}}([\{m_i | i \in A\}])$$

we can obtain short fragments. Indeed, in this case we obtain that $F_i < m_i \leq max(m_1, \ldots, m_n) < S^{\frac{1}{\alpha}}$ and, thus[2], $|F_i| < \frac{|S|}{\alpha}$, for all $1 \leq i \leq n$.

3 Computational Secure Secret Sharing

A secret sharing scheme starts with a *secret* and then derives from it certain *shares* (*shadows*). The secret may be recovered only in the case of possessing a certain predetermined set of shares. The initial applications of secret sharing were safeguarding cryptographic keys and providing shared access to strategical resources. Threshold cryptography (see, for example, [8]) and some e-voting schemes (see, for example, [7]) are more recent applications of the secret sharing schemes.

In the first secret sharing schemes only the number of shares was important for recovering the secret. Such schemes have been referred to as *threshold* secret sharing schemes. We mention Shamir's threshold secret sharing scheme [19] based on polynomial interpolation, Blakley's geometric threshold secret sharing scheme [5], Mignotte's threshold secret sharing scheme [14] and Asmuth-Bloom threshold secret sharing scheme [1], both based on the Chinese remainder theorem. Ito, Saito, and Nishizeki [12], Benaloh and Leichter [4] give constructions for more general secret sharing schemes.

[2]For a positive integer x, $|x| = \lceil log_2(x+1) \rceil$.

Definition 4. *Let n be an integer, $n \geq 2$ and $\mathcal{A} \subseteq \mathcal{P}(\{1,2,\ldots,n\})$. An* $\mathcal{A}$-secret sharing scheme *is a method of generating $(S,(I_1,\ldots,I_n))$ such that*

- *for any $A \in \mathcal{A}$, the problem of finding the element S, given the set $\{I_i \mid i \in A\}$ is "easy";*
- *for any $A \in \mathcal{P}(\{1,2,\ldots,n\}) \setminus \mathcal{A}$, the problem of finding the element S, given the set $\{I_i \mid i \in A\}$ is intractable.*

The set $\mathcal{A}$ will be referred to as the *authorized access structure* or simply as the *access structure*, S will be referred to as the *secret* and $I_1,\ldots,I_n$ will be referred to as the *shares* (or the *shadows*) of S. The elements of the set $\mathcal{A}$ will be referred to as the *authorized access sets* of the scheme.

As it can be seen from the above definition, the difference between information dispersal and secret sharing is that in the case of information dispersal there is no restriction on the unauthorized sets.

In a *perfect* secret sharing scheme, the shares of any unauthorized set give no information (in information-theoretical sense) about the secret. Krawczyk [13] proposed combining perfect threshold secret sharing schemes with encryption and information dispersal in order to construct non-perfect threshold secret sharing schemes with smaller shares, maintaining, in the same time, a reasonable level of security. Béguin and Cresti [2] have remarked that this technique can be easily extended to more general access structures. We present next the technique for the general case.

The generation of shares

- Choose an encryption function $e_{\{\}}$ and generate a random key K for it;
- Compute $\overline{S} = e_K(S)$;
- Use an $\mathcal{A}$-information dispersal scheme to partition the information $\overline{S}$ into n fragments $F_1,\ldots,F_n$;
- Use a perfect $\mathcal{A}$-secret sharing scheme to construct the shares $K_1,\ldots K_n$ corresponding to the secret K;
- The shares I_i are chosen as $I_i = (F_i,K_i)$, for all $1 \leq i \leq n$.

The reconstruction of the secret

- Having the shares $\{I_i | i \in A\}$, for some minimal authorized set A, the secret S can be recovered as follows:
  - The information $\overline{S}$ is recovered using the reconstruction algorithm applied to $\{F_i | i \in A\}$;
  - The key K is recovered using the reconstruction algorithm applied to $\{K_i | i \in A\}$;
  - The secret S is recovered as[3] $S = d_K(\overline{S})$.

4 Conclusions

We have presented a new general information dispersal scheme based on the Chinese remainder theorem. We have also indicated how to select the parameters of the scheme in order to obtain short fragments. Finally, we have presented the application of information dispersal in secret sharing.

An interesting open problem is to efficiently generate $m_1,\ldots,m_n$, and the biggest $\alpha \geq 1$ such that

$$(max(m_1,\ldots,m_n))^{\alpha} < min_{A \in \mathcal{A}_{min}}([\{m_i | i \in A\}]).$$

We shall consider this problem in our future work.

Acknowledgements Research reported here was partially supported by the National University Research Council of Romania under the grant CNCSIS632/2005.

[3] $d_{\{\}}$ represents the decryption function corresponding to $e_{\{\}}$, i.e., the function that satisfies $d_K(e_K(S)) = S$, for any key K and any plaintext x.

References

[1] C. A. Asmuth and J. Bloom. A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, IT-29(2):208–210, 1983.

[2] P. Béguin and A. Cresti. General short computational secret sharing schemes. In L. C. Guillou and Quisquater J.-J, editors, *Advances in Cryptology - EUROCRYPT '95,*, volume 921 of *Lecture Notes in Computer Science*, pages 194–208, 1995.

[3] P. Béguin and A. Cresti. General information dispersal algorithms. *Theoretical Computer Science*, 209(1-2):87–105, 1998.

[4] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advanced in Cryptology-CRYPTO' 88*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer-Verlag, 1989.

[5] G. R. Blakley. Safeguarding cryptographic keys. In *National Computer Conference, 1979*, volume 48 of *American Federation of Information Processing Societies Proceedings*, pages 313–317, 1979.

[6] H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer-Verlag, 4th edition, 2000.

[7] R. Cramer, M. K. Franklin, B. Schoenmakers, and M. Yung. Multi-authority secret-ballot elections with linear work. In U. Maurer, editor, *Advances in Cryptology - EuroCrypt '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 72–83. Springer-Verlag, 1996.

[8] Y. Desmedt. Some recent research aspects of threshold cryptography. In E. Okamoto, G. I. Davida, and M. Mambo, editors, *ISW '97: Proceedings of the First International Workshop on Information Security*, volume 1396 of *Lecture Notes in Computer Science*, pages 158–173. Springer-Verlag, 1998.

[9] C. Ding, D. Pei, and A. Salomaa. *Chinese remainder theorem: applications in computing, coding, cryptography*. World Scientific Publishing Co., Inc., 1996.

[10] A. S. Fraenkel. New proof of the generalized Chinese remainder theorem. *Proceedings of American Mathematical Society*, 14:790–791, 1963.

[11] H. Garner. The residue number system. *IRE Transactions on Electronic Computers*, EC-8:140–147, 1959.

[12] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. In *Proceedings of the IEEE Global Telecommunications Conference, Globecom '87*, pages 99–102. IEEE Press, 1987.

[13] H. Krawczyk. Secret sharing made short. In D. R. Stinson, editor, *Advances in cryptology -CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[14] M. Mignotte. How to share a secret. In T. Beth, editor, *Cryptography-Proceedings of the Workshop on Cryptography, Burg Feuerstein, 1982*, volume 149 of *Lecture Notes in Computer Science*, pages 371–375. Springer-Verlag, 1983.

[15] O. Ore. The general Chinese remainder theorem. *American Mathematical Monthly*, 59:365–370, 1952.

[16] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24:106–110, 1978.

[17] J.-J. Quisquater and C. Couvreur. Fast decipherment algorithm for the RSA public-key cryptosystem. *IEE Electronics Letters*, 18 (21):905–907, 1982.

[18] M. O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of ACM*, 36(2):335–348, 1989.

[19] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

Sorin Iftene
Faculty of Computer Science
"Al. I. Cuza" University
Iaşi, Romania
E-mail: siftene@infoiasi.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 280-283

Computational Aspects in Excitable Media. The Case of Vortex Phenomena

Adela Ionescu, Mihai Costescu

Abstract: The turbulence phenomenon is a modern component of fluid kinematics. In this area, the turbulent mixing is distinguished by its importance in applied engineering (including technical, social, economic applications). The turbulent mixing is an important feature of far from equilibrium models. Studying a mixing for a flow implies the analysis of successive stretching and folding phenomena for its particles, the influence of parameters and initial conditions, and also the issue of significant events - such as rare events - and their physical mean. A comparison between two and three dimensional flows produces useful remarks.

Keywords: tendril-whorl flow, mixing, rotation, stretching

1 Introduction

The turbulence mathematical term can be defined as "chaotic behavior of far from equilibrium systems, with very few freedom degrees". In this area two important theories are distinguished:

a) The transition theory from smooth laminar flows to chaotic flows, characteristic to turbulence.

b) Statistic studies of the complete turbulent systems.

The statistical idea of flow is represented by the map:

(1) $x = \Phi_t(X),\ with\ X = \Phi_t(t=0)(X)$

We say that X is mapped in x after a time t. In the continuum mechanics the relation (1) is named *flow*, and it is a diffeomorphism of class C^k.

Moreover, (1) must satisfy the relation:

(2) $0 \langle J \langle \infty, J = \det\left(\frac{\partial x_i}{\partial X_j}\right)$, or $J = det(D\Phi_t(X))$,

where D denotes the derivation with respect to the reference configuration, in this case X. The relation (2) implies two particles, X_1 andX_2, which occupy the same position x at a moment. Non-topological behavior (like break up, for example) *is not allowed*.

With respect to X there is defined the basic measure of deformation, the deformation gradient, F, namely:

(3) $F = (\nabla_X \Phi_t(\mathbf{X}))^T,\ F_{ij} = \left(\frac{\partial x_i}{\partial X_j}\right)$,

where ∇_X denotes differentiation with respect to X. The basic measure for the deformation with respect to x is the velocity gradient.

After defining the basic deformation of a material filament and the corresponding relation for the area of an infinitesimal material surface [1,3], we can define the basic deformation measures: the length deformation λ and surface deformation η, with the relations [1,3]:

(4) $\lambda = (C : MM)^{\frac{1}{2}},\ \eta = (\det F)\cdot\left(C^{-1} : NN\right)^{\frac{1}{2}}$,

with $C(= F^T \cdot F)$ the Cauchy-Green deformation tensor, and the vectors M, N are the orientation versors in length and surface respectively. The scalar form for (4), used in practice, is:

(5) $\lambda^2 = C_{ij}\cdot M_i \cdot N_j,\ \eta^2 = (\det F)\cdot\left(C_{ij}^{-1}\cdot N_i \cdot N_j\right),\ with\ \sum M_i^2 = 1,\ \sum N_j^2 = 1.$

The deformation tensor F and the associated tensors C, C^{-1} represent the basic quantities in the deformation analysis for the infinitesimal elements.

In the above framework the mixing concept implies the *stretching* and *folding* of the material elements. If in an initial location P there is a material filament dX and an area element dA, the specific length and surface deformations are given by the relations [1,3]:

(6) $\frac{D(\ln\lambda)}{Dt} = D : mm,\ \frac{D(\ln\eta)}{Dt} = \nabla v - D : nn$

where D is the deformation tensor, obtained by decomposing the velocity gradient in its symmetric and non-symmetric part [3].

We say that the flow $x = \Phi_t(X)$ *has a good mixing* if the mean values $D(ln\lambda)/Dt$ and $D(ln\eta)/Dt$ are not decreasing to zero, for any initial position P and any initial orientations M and N. As the above two quantities are bounded, the deformation efficiency can be naturally quantified. Thus, there is defined [3] the *deformation efficiency in length*, $e_\lambda = e_\lambda(X,M,t)$ of the material element dX, as:

(7) $e_\lambda = \frac{D(ln\lambda)/Dt}{(\mathbf{D}:\mathbf{D})^{1/2}} \leq 1,$

and similarly, the *deformation efficiency in surface*, $e_\eta = e_\eta(X,N,t)$ of the area element dA: in the case of an isochoric flow (the jacobian equal 1), we have:

(8) $e_\eta = \frac{D(ln\eta)/Dt}{(\mathbf{D}:\mathbf{D})^{1/2}} \leq 1.$

2 The tendril-whorl flow. Present aims

2.1 Recent results

As shown in [2,3], two-dimensional flows increase their length by forming two basic kinds of structures: tendrils and whorls and their combinations. The tendril-whorl flow (TW) introduced by Khakhar, Rising and Ottino (1987) is a discontinuous succession of extensional flows and twist maps. Even the simplest case is complex enough. The physical motivation for this flow is that locally, a velocity field can be decomposed into extension and rotation.

In the simplest case of the TW model, the velocity field over a single period is given by its extensional part:

$(9)_1$ $v_x = -\varepsilon \cdot x,$
$v_y = \varepsilon \cdot y, \quad 0 < t < T_{ext}$

and its rotational part:

$(9)_2$ $v_r = 0,$
$v_\theta = -\omega(r),\ T_{ext} < t < T_{ext} + T_{rot},$

where T_{ext} denotes the duration of the extensional component and T_{rot} the duration of rotational component.

The model consists of vortices producing whorls which are periodically squeezed by the hyperbolic flow leading to the formation of tendrils, and the process repeats. The function $\omega(r)$ is positive and specifies the rate of rotation.

In [2] it was studied the extensional part, the model $(9)_1$ of TW model. For the solution of this model,

(10) $x = X \cdot \exp(-\varepsilon \cdot T_{ext}),$
$y = Y \cdot \exp(\varepsilon \cdot T_{ext})$

the gradient deformation F and the Cauchy-Green tensors C, C^{-1} had quite simple forms, and therefore the deformations in length and surface λ^2 and η^2 were appropriate [2]. It was found that the expressions of the deformations in length and surface are quite similar. Therefore it is important to calculate and compare the deformation efficiencies e_λ and e_η. Their expressions are the following [2]:

(11) $e_\lambda = 2\varepsilon \cdot \left(1 - \frac{2\exp(-2\varepsilon T_{ext}) \cdot M_1^2}{\exp(-2\varepsilon T_{ext}) \cdot M_1^2 + \exp(2\varepsilon T_{ext}) \cdot M_2^2}\right)$

(12) $e_\eta = 2\varepsilon \cdot \left(1 - \frac{2\exp(-2\varepsilon T_{ext}) \cdot N_2^2}{\exp(-2\varepsilon T_{ext}) \cdot N_2^2 + \exp(2\varepsilon T_{ext}) \cdot N_1^2}\right)$

where $M_1^2 + M_2^2 = 1$, $N_1^2 + N_2^2 = 1$.

2.2 Graphic analysis

For the beginning there were considered some irrational – but equal- values both for the length and surface versors. For several moments of T_{ext} (the duration of extensional component), there were analyzed the behavior of e_λ and e_η as parametric functions of time. It was found [2] that both functions have a nonlinear behavior.

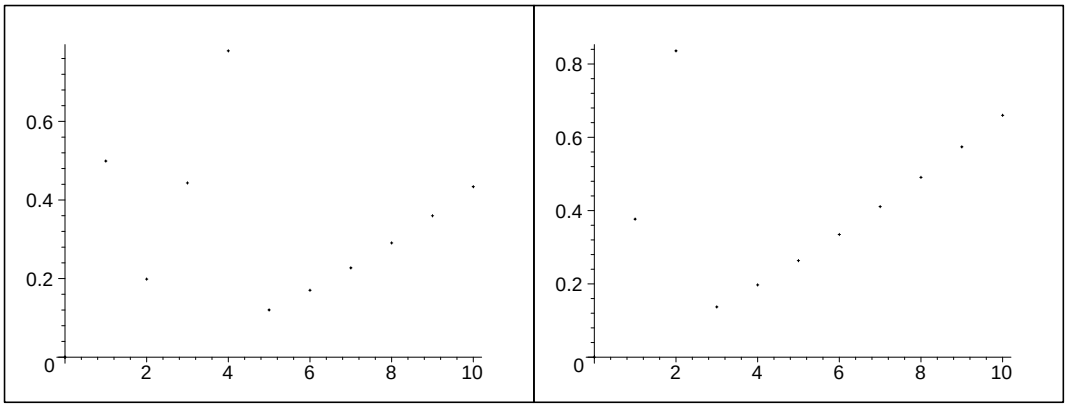
For better outline this, a graphic analysis was realized. For the versor values, there were considered three sets of values, as following:

(a) $(M_1,M_2) = (N_1,N_2) = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$; (b) $(M_1,M_2) = (N_1,N_2) = \left(\frac{1}{\sqrt{3}}, \frac{\sqrt{2}}{\sqrt{3}}\right)$; (c) $(M_1,M_2) = (N_1,N_2) = \left(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\right)$.

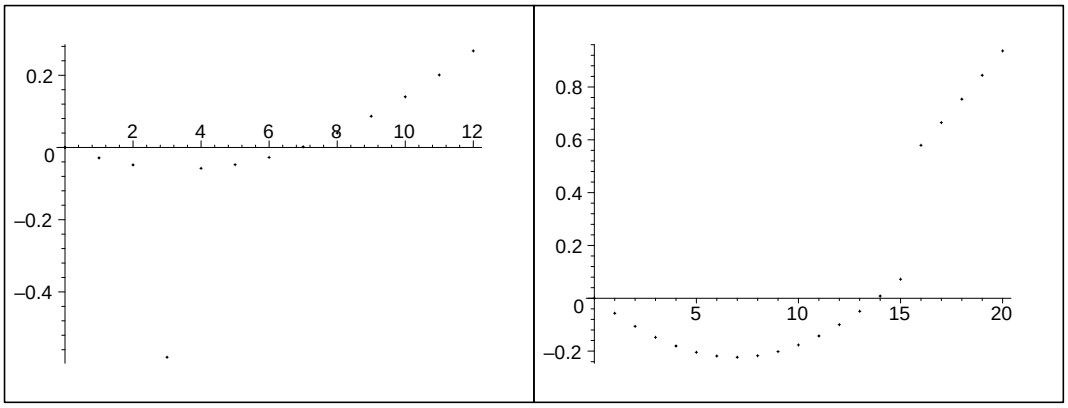
For the parameter $0\langle\varepsilon\langle 1$ there were considered two values: $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.08$. Thus, the following situations were identified:

(a1) the case (a) with ε_1; (a2) the case (a) with ε_2;
(b1) the case (b) with ε_1; (b2) the case (b) with ε_2;
(c1) the case (c) with ε_1; (c2) the case (c) with ε_2.

For each of the six above cases, it was calculated e_λ and e_η , following the formulas (13) and (14) respectively. Let us remember that the deformation efficiencies e_λ and e_η are differential equations, since they are calculated by the relations [1,4]:



(a) Picture 1 (b) Picture 2



(c) Picture 1 (d) Picture 2

$e_\lambda = \frac{1}{2\lambda^2} \cdot \frac{d\lambda^2}{dt}, e_\eta = \frac{1}{2\eta^2} \cdot \frac{d\eta^2}{dt}$

The differential equations are as following:

$(a_1)\ e_\lambda = e_\eta = 0.1 \cdot \left(1 - \frac{2 \cdot \exp(-0.1 \cdot T)}{\exp(-0.1 \cdot T) + \exp(0.1 \cdot T)}\right)$; $(a_2)\ e_\lambda = e_\eta = 1.6 \cdot \left(1 - \frac{2 \cdot \exp(-1.6 \cdot T)}{\exp(-1.6 \cdot T) + \exp(1.6 \cdot T)}\right)$;

$(b_1)\ e_\lambda = 0.1 \cdot \left(1 - \frac{\frac{2}{3} \cdot \exp(-0.1 \cdot T)}{\frac{1}{3} \cdot \exp(-0.1 \cdot T) + \frac{2}{3} \cdot \exp(0.1 \cdot T)}\right); e_\eta = 0.1 \cdot \left(1 - \frac{\frac{4}{3} \cdot \cdot \exp(-0.1 \cdot T)}{\frac{1}{3} \cdot \exp(0.1 \cdot T) + \frac{2}{3} \cdot \exp(-0.1 \cdot T)}\right)$;

$(b_2)\ e_\lambda = 1.6 \cdot \left(1 - \frac{\frac{2}{3} \cdot \exp(-1.6 \cdot T)}{\frac{1}{3} \exp(-1.6 \cdot T) + \frac{2}{3} \exp(1.6 \cdot T)}\right)$; $e_\eta = 1.6 \cdot \left(1 - \frac{\frac{4}{3} \cdot \exp(-1.6 \cdot T)}{\frac{1}{3} \exp(1.6 \cdot T) + \frac{2}{3} \exp(-1.6 \cdot T)}\right)$;

$(c_1)\ e_\lambda = 0.1 \cdot \left(1 - \frac{\frac{2}{5} \cdot \exp(-0.1 \cdot T)}{\frac{1}{5} \cdot \exp(-0.1 \cdot T) + \frac{4}{5} \cdot \exp(0.1 \cdot T)}\right)$; $e_\eta = 0.1 \cdot \left(1 - \frac{\frac{8}{5} \cdot \cdot \exp(-0.1 \cdot T)}{\frac{1}{5} \cdot \exp(0.1 \cdot T) + \frac{4}{5} \cdot \exp(-0.1 \cdot T)}\right)$;

$(c_2)\ e_\lambda = 1.6 \cdot \left(1 - \frac{\frac{2}{5} \cdot \exp(-1.6 \cdot T)}{\frac{1}{5} \exp(-1.6 \cdot T) + \frac{4}{5} \exp(1.6 \cdot T)}\right)$; $e_\eta = 1.6 \cdot \left(1 - \frac{\frac{8}{5} \cdot \exp(-1.6 \cdot T)}{\frac{1}{5} \exp(1.6 \cdot T) + \frac{4}{5} \exp(-1.6 \cdot T)}\right)$.

T denotes the duration of extensional component, i.e. $T_{ext} := T$.

For solving these differential equations, it was applied the Maple numeric procedure Dsolve [6], obtaining in each case a list of pairs . The index i goes from 1 to 15, for the moment, as we are not interested to give very few values for the period T_{ext}. Only few cases contain more time period, where it was necessary to outline the nonlinearity.

Finally, using the plot lists there were realized discrete plots with the Maple function Pointplot [6]. Between the ten plots obtained, four were nonlinear. We shall expose these four plots. The semnification order is the following: Picture1 - case a1 with $e_\lambda = e_\eta$; Picture2- case b1 with e_λ; Picture3 - case b2 with e_η; Picture4 - case c1 with e_η.

3 Remarks

As one can see in the pictures, in the case (b) - for ε_1 the deformation efficiency in length e_λ has a nonlinear behavior and for ε_2 the deformation efficiency in surface e_η is negative nonlinear. In the case (c) only for ε_1 we have e_η negative nonlinear. Thus, some remarks are useful for immediate aims:

1) Almost one half of cases present nonlinearities. That's why we can say that the periodic flows, even with a simpler mathematical model like (11), can have a nonlinear behavior.

2) It would be useful to search rare events for this model. In [1], the rare events were defined as the events of breaking up of filaments of the aquatic algae exposed to the vortex phenomena [5]. The conclusion was that the turbulent mixing for a 3D (no periodic!) flow can be associated to a far from equilibrium phenomena. The problem is if the same thing can happen with periodic flows. This is a proximal aim.

3) The parameter T_{ext} can be measured in seconds, minutes or even in larger units, depending on the context. The same fact can be found also in [1] for three dimensional flow, where the turbulence occurs at small values of the time units, being in agreement with experiments. Therefore an analysis for larger T_{ext} would be useful.

4) These conclusions are preliminary, as we have not exhausted all irrational cases for the length and surface versors, all situations for the time period T_{ext} . A complete analysis for both deformations efficiencies will come soon. Comparing to [1], this will be possible taking into account more statistical cases.

References

[1] A. Ionescu, "The structural stability of biological oscillators. Analytical contributions", *Ph.D. Thesis*. Politechnic, University of Bucarest, 2002.

[2] A. Ionescu, "The qualitative study of the mixing for the tendril-whorl flow", *Proceedings of the International Conference ICEEMS2005* (International Conference on Economic Engineering and Manufacturing Systems), 20-22.10.2005, Alba-Iulia.

[3] J.M Ottino, "The kinematics of mixing: stretching, chaos and transport", *Cambridge University Press*. 1989

[4] J.M. Ottino, W.E. Ranz & C.W. Macosko. "A framework for the mechanical mixing of fluids", *AlChE J*. 27, 565-577, 1981.

[5] St.N. Savulescu, "A special vortex tube for particle processing flows", *Rev. Roum. Sci. Techn.-Mec. Appl.*, Tome43, no5, 1998, 611-615.

[6] R. S. Stepleman, Alan C. Hindmarsh (eds.), "Odepack, a Systemized Collection of ODE Solvers", *North-Holland*, Amsterdam, 1983.

Adela Ionescu, Mihai Costescu
University of Craiova, Romania
E-mail: adela0404@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 284-289

Multimedia Educational Software for Producing Graphs of Mathematical Functions

Anca Elena Iordan, Manuela Pănoiu

Abstract: The informative society needs important changes in educational programs. The informational techniques needs a reconsideration of the learning process, of the programs, manuals structures, a reconsideration of the methods and organization forms of the didactic activities, taking into account the computer assisted instruction and self instruction. This paper presents a software package, which can be used as educational software. This paper present a graphical user interface implement in C++ Builder useful for producing graphs of mathematical functions.

Keywords: Educational software, grammar, backtrack parsing, function.

1 Introduction

In the condition of informatics society whose principal source in the social-economic development is to produce and consumption the information, the complex and fast knowledge of the reality for rational, opportune, effective decisions is a desideratum which generate the necessity to form some superior level habituation in information manage for the whole population.

The computers and their programs offer to the users powerful capabilities for the information manipulation:
- Image and text visualize on the screen which can be manipulate later;
- Memory storage of an important quantity of information, his accessing and selection of a part of them;
- Possibility to realize a great volume of computation;
- Possibility of equipment control and fast decisions;
- Computer based training.

This facilities offer to the microcomputers higher educational capabilities versus other technologies used in education and provide learning controlled based on many parameters: intellectual aptitude, level of knowledge, abilities, rhythm of work.

2 Computer based training as a didactic method

The informatics society makes sensitive modification in education programs. In this scope, the school must prepare programmers, maintenance technicians, etc. In the same time it is necessary that the teacher make ready to use the computer in education process.

These informational techniques impose to reorganize the contents of the education process, of the programs, course books and manuals, to reconsider the methods and organization forms of didactic activities, which follow to be center on individualization of the teaching process. As a method of the informational didactic, the computer-based training is based on the programmed teaching. N. Crowder work out a new programming type: the branch programming which is characterized by: division of the content in small steps, his successive presentation according to the student needs and corrective feedback, use of author language. The programmed teaching consist in information presentation in small units, logic structured, units that compose a program, the teaching program.

The user will have possibility that after each sequence to have a knowledge about the measure of understanding the give information. The programmed teaching method organize the didactic action applying the cybernetic principles to the teaching-learning-evaluating activities level, considering like a complex and dynamic system, composed as an elements ensemble and inter-relations and develop his personal principles valid on the strategic level in any cybernetic organization form of teaching.

On the other hand, programmed teaching assume some principles which the teaching program must respect:
- The small steps principle consists in progressive penetration, from simple to complex, in a subject content which logic divided in simple units series lead to minimal knowledge, which later will form an ensemble. This principle regards the subject division in contents/information units that give to user the chance to succeed in his teaching activities;
- The principle of personal rhythm of study regard mannerism observance and capitalization of each user of the

program which will be able to make the sequences of knowledge learning or control, in a personal rhythm appropriate to his psycho-intellectual development, without time limits. The user can progress in the program only if he accomplished the respective sequence requirement;
- The active participation principle, or active behavior, regard user effort trend into selection, understanding and applying the necessary information in elaboration of a correct answer. On each step the user is liable to an active participation to resolve the step job;
- The principle of inverse connection, regard positive or negative inputs of user competence, refer to the success or breakdown in task performed;
- The immediate and directly control of the task work precision with the possibility to progression to the next sequence, in case of success;
- The repetition principle, based to the fact that the programs are based on return to the users initial knowledge.

The combined programming interposes the linear and branch sequence according to teaching necessities. After linear and branch programming the computer aided generative teaching has appear, where the exercises are gradually present, with different difficulty steps and answers on the students questions. The expert system consists of self-teaching training programs, tutorial strategies, and the usage of natural language, mixed initiative and some complex representation of knowledge usage. The simulation is a training computer programs onset characterized by the fact that the computer is like a lab witch contains typical interactive graphical programs.

The computer based programmed teaching realize learning process with a inputs flow - the command, an executive controlled system, an output flux - control and a control system functions which correct measure establish. In such a system have tree stages of teacher perceive: teaching, evaluating and the feedback loop closing, the computer being present in all of tree stages.

3 Application present

The application is implemented in Borland C++ Builder 6.0, under Microsoft Windows operating system. Consider the following grammar that generates mathematical expressions:
The axiom: S
The terminal symbols: $(,),.,+,-,*,/,\wedge,0,1,2,3,4,5,6,7,8,9,x,s,i,n,c,o,t,g,a,h,l,e,p,b,r,d$
The nonterminal symbols: S, E, T, F, N, I, R, A, M, C
The rules :

S $\longrightarrow$ E
E $\longrightarrow$ T | TAE
T $\longrightarrow$ F | FMT
F $\longrightarrow$ x | (E) | sin(E) |cos(E) | tg(E) | ctg(E) | asin(E) | acos(E) | atg(E) | actg(E)
F $\longrightarrow$ sh (E) | ch(E) | th(E) | cth(E) | ash(E) | ach(E) | ath(E) | ach(E)
F $\longrightarrow$ log(E) | ln(E) | lg(E) | exp(E) | abs(E) | rad(E) | N
N $\longrightarrow$ I.R
I $\longrightarrow$ CI | C
R $\longrightarrow$ CR | C
C $\longrightarrow$ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
A $\longrightarrow$ + | -
M $\longrightarrow$ $*$ | / | $\wedge$

The implementation of the grammar can be as follows:

```
typedef struct { char* m;
                 n[5];} productie;
class gramatica { private:
                      char a;
                      char* t;
                      char* n;
                      productie p[35];
                      int nr;
                  public:
                      gramatica(char* nume-fisier);
                      ~gramatica(){ delete [] t;delete [] n;}
                      char* terminale();
```

```
int nr-productii(){ return nr;}
char axioma(){return a;}
char* prod1(int i){ return p[i].m;}
char* prod2(int i){ return p[i].n;}
};
```

Parsing is the process of analyzing an input sequence in order to determine its grammatical structure with respect to a given formal grammar. The parsing state is: (s, i, α, β), where:

- **s** is the state which can be: - q (normal state);
  - b (back state);
  - e (error state);
  - t (true state);
- **i** is index in the input string;
- α is the sequence of recognized symbols and the index of used rules;
- β is the sequence of symbols which were not yet recognized.

The analyse algorithm include the next states:

- initial state: (q, 1, ε, S);
- expansion: (q, j, α, Aβ) $\longrightarrow$ (q, j, αA_1, $\alpha_1\beta$), where A $\longrightarrow$ α_1 is the first rule of A;
- advance: (q, j, α, aβ) $\longrightarrow$ (q, j+1, αa, β), when in the top of stack is a terminal symbol which is identical with the current symbol of input string;
- noncoincidence: (q, j, α, iβ) $\longrightarrow$ (b, j, α, iβ), when in the top of stack is a terminal symbol which is not identical with the current symbol of input string;
- return:
  1. (b,j, αi, β) $\longrightarrow$ (b, j-1, α, iβ), when in the top of stack is a terminal symbol;
  2. (b,j, αA_i, $\alpha_i\beta$) $\longrightarrow$ (q, j, αA_{i+1}, $\alpha_{i+1}\beta$);
  3. (b,j, αA_n, $\alpha_n\beta$) $\longrightarrow$ (b, j, α, Aβ);
- failure: (b, j, α, Sβ) $\longrightarrow$ (e, j, α, Sβ);
- success: (q, n+1, α, ε) $\longrightarrow$ (t, n+1, α, ε).

The backtrack parsing for the language generated by the grammar can be implemented as follows:

```
typedef struct { char s;
                 short i;
                 char a[300];
                 char b[30];} stare;
class automat { private:
                  gramatica g;
                  stare s;
                  char* w;
                public:
                  automat(gramatica g1, char* w1);
                  ~automat(){delete [] w;}
                  stare initializare();
                  stare expandare();
                  stare avans();
                  stare necoincidenta();
                  stare revenire1();
                  stare revenire2();
                  stare revenire3();
                  stare esec();
                  stare succes();
                  void afis();
                  int exista();
                  int terminal(char c);
                  int algoritm(); };
```

The main application window allows, by a menu, to select an option. The "IESIRE" option is use for closing window.

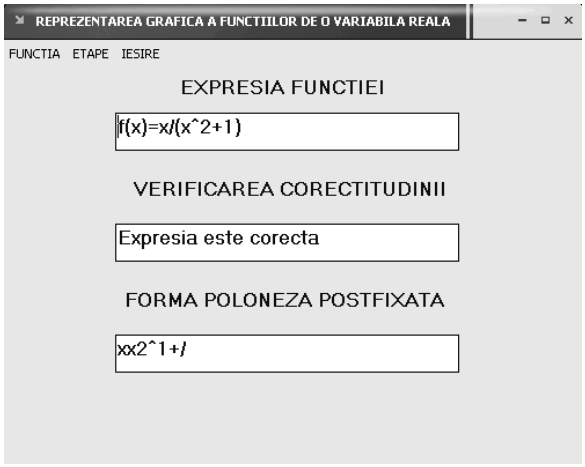


Figure 1: The main window

The "FUNCTIA" option allows verification of correctness of input sequence and determination of reverse polish notation. For example, if the input string is "x/(x∧2+1)" then the final state of parsing is the true state, so the function expression is correct (Figure 2). In this notation the above expression would be: xx2∧1+/.

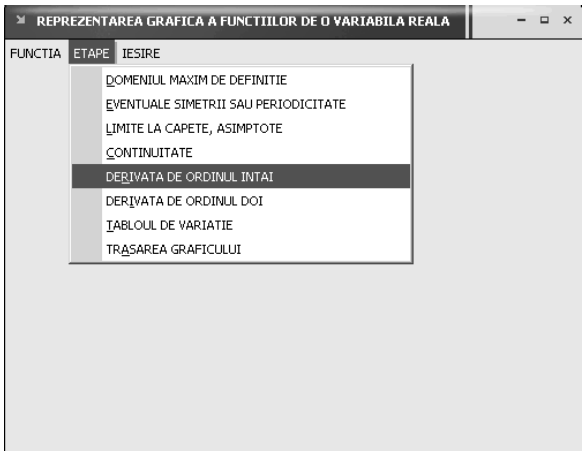


Figure 2: The postfix polish notation

If x is 3 and reading from left to right, this is interpreted as follows:
• Push 3 onto the stack.
• Push 3 onto the stack. The stack now contains (3, 3).
• Push 2 onto the stack. The stack now contains (3, 3, 2).
• Apply the ∧ operation: take the top two numbers off the stack and put the result back on the stack. The stack now contains (3, 9).
• Push 1 onto the stack. It now contains (3, 9, 1). " Apply the + operation: take the top two numbers off the stack, add them together and put the result back on the stack. The stack now contains (3, 10).
• Apply the / operation: take the top two numbers off the stack and put the result back on the stack. The stack now contains just the number 0.3.

The "ETAPE" option allows crossing of necessary steps for graphic representation of function:
• The maximum domain for definition of function;
• Eventual symmetry or parity;
• Asymptotes;
• Continuity of the function;
• The first differential of the function; Critical points;
• The second differential of the function; Modulation points;

• Variation table;
• The graph drawing of the function.

The "DERIVATA DE ORDINUL INTAI" option allows determination of the first differential of the function and the critical points. For example, if the expression of the function is ”x/(x∧2+1)” then the first differential of the function is ”(-x∧2+1)/(x∧2+1)∧2” and the critical points are: x1=-1, x2=1(Figure 3).

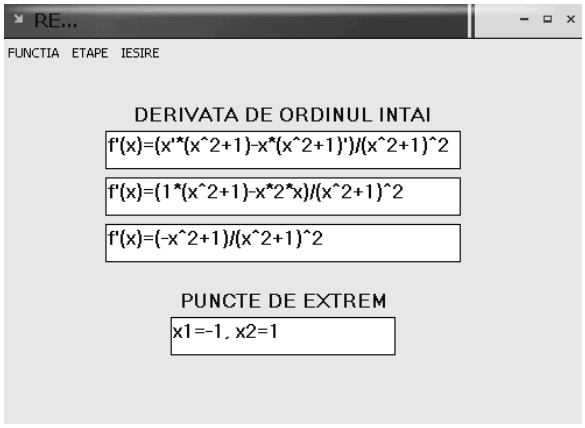


Figure 3: The first differential of the function

The "TRASAREA GRAFICULUI" option allows drawing of function graph. For previous function, the graph is represent from figure 4.

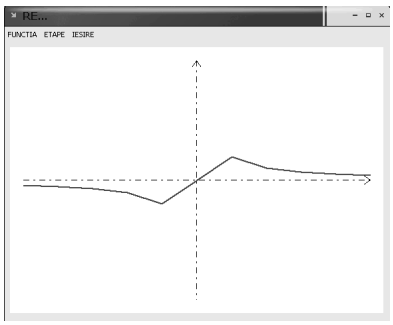


Figure 4: The graph of the function

4 Summary and Conclusions

On this application, authors take into consideration the condition, which must accomplish a courseware, being made necessary steps. So, in elaboration and utilization of this application must take into consideration next criteria:

- To follow up the curriculum for a specific domain;
- To accomplish some teaching and learning strategy. In this kind of self-instruction and evaluation program it must find basic notions and representation and scanning notions. Animation and graphical modeling must represent the graphical construction way and also scanning of them;
- To exist the possibility to use parameterized variable, in conditions in which users have the possibility to input the variables value;
- To present a method in which the user can be informed about how can use graphical module, i.e. an interaction user-computer exist.

The presented application accomplishes these criteria, and for this we consider that is a good example of how educational software must be realized.

References

[1] Anne Mcdougall, David Squires, ”Empirical study of a new paradigm for choosing educational software”, Computer Education, vol 25, no 3, Elsevier Science, 1995

[2] Glenn W. Rowe, Peter Gregor,”A computer based learning system for teaching computing: implementation and evaluatio”, Computer and Education , Elsevier Science, 1999

[3] Knuth D. E.,”Fundamental Algorithm”, Third Edition, Massachusetts

[4] Aho,A.V., Ullman,J.D.”The Theory of Parsing, Translation, and Compilin”, Vol.1. Englewood Cliffs, N.J.:Prentice-Hall, 1972

[5] Orman, G. ”Limbaje formale”, Editura Tehnicã, Bucuresti, 1982.

[6] Paun,Gh. ”Gramatici contextuale”, Editura Academiei RSR, Bucuresti,1982.

[7] Serbanati, L.”Limbaje de programare si compilatoare”, Editura Academiei, Bucuresti, 1987

[8] Mateescu, A. ”Structuri matematice discrete. Aplicati”, Editura Academiei, Bucuresti, 1989

[9] Simovici, D. ”Limbaje formale si tehnici de compilar”, Editura didactica si pedagogica, Bucuresti, 1978

Anca Elena Iordan , Manuela Pănoiu
Technical University of Timisoara
Engineering Faculty of Hunedoara
Address: Revolutiei str, no. 5, cod 331115, Hunedoara, Romania
E-mail: anca.iordan@email.ro, m.panoiu@fih.upt.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 290-294

Speech Recogniton Using Quantum Signal Processing

S. Karthikeyan, S. Sasikumar

Abstract: Speech processing by computer is a major field of endeavor. Speech is one of the natural means of exchanging information for human. The system verifies the feature extraction for speech recognition system based on Eigen properties and some quantization principles. It is used to find the improvement in the quality of speech recognition and to provide an efficient and accurate mechanism to describe human speech to text. In my Paper a new approach, QSP is proposed in which the goal of the study is to improve the accuracy. QSP stands for Quantum Signal Processing. It is related to quantum mechanics. The mathematical structure of quantum mechanics is connected to physical reality through the concept of measurement. This approach assumes that improving the accuracy will necessarily result in improved recognition performance and ignores the manner in which the isolated word recognition systems operate.

Keywords: quantum mechanics, concept of measurement, inner product structure, least square, Eigen decomposition

1 Introduction

Prior to the introduction of vector quantization based word recognizer, speech recognition was based almost entirely on Dynamic Time Warping (DTW) techniques. These DTW recognizers are limited in that they are speaker dependent and can operate only on discrete words or phrases. The recognition technique employed in DTW is straight forward a set of speech templates are maintained in memory for each word in the vocabulary. These templates are based on linear predictive coding models. As a new word or phoneme is acquired and placed in the speech queue, its features or characteristics are compared to each memory resident template one at a time. As the acquired speech frame is compared, it is stretched or compressed in time to optimize the correlation between the memory - resident templates and queued speech frame. As this warping process progresses, the result of the optimized correlation is logged and the score is updated. This repeated for each template in the current vocabulary list (Paw ate Etal, 1993).

A speaker independent, continuous recognition is possible using vector quantization. The basic concepts are taken from the University of Leeds website explained using the weather prediction examples. The whole organization of this project is based on the Texas Instruments, Application manual (Paw ate Etal, 1993). The mathematical techniques and procedure for implementing the algorithms are referred from the IEEE paper (Rabiner, 1989). The training algorithm is based on the (Jurafsky, 2002) and (Steve Young, 2001).

Speech recognition is fields in which theory can be applied to real systems to a great extend through computers/microcomputers. The complexity of the theory behind speech recognition applications may prevent us to grasp every detail of the speech recognition methods; however there are a number of useful resources which can help us to determine the most successful algorithms. Implementing speech recognition using a digital signal processor is a further goal which is beyond the scope of this paper. However, the high-level programming language implementation will be realized taking the limitations of real-time signal processing into account .Additional resources, either cited below or to be found later such as the articles in IEEE's various journals about signal and/or speech processing, may help us fine tune the speech recognition system.

2 Implementation of QSP

One of the fast emerging fields in electronics is Quantum signal processing .A program for caring out a research on the implementation of qsp. In our project, a new approach qsp is proposed in which the goal of signal processing to generate a sequence of features. QSP stands for quantum signal processing and it is related to quantum mechanics. This approach assumes that improving the quality of speech waveform will necessarily result in improved recognition performances and ignores the manner in which the speech recognition systems operate. QSP is aimed at developing a new or modifying existing signal processing algorithm by borrowing from the principle of quantum mechanics and some of its axioms and constraints. However, in contrast to such fields as quantum

computing and quantum information theory, it does not inherently depend on the physics associated with quantum mechanics.

3 Eigen Value of Matrices

Computing eigenvalues of matrices

Suppose that we want to compute the eigenvalues of a given matrix. If the matrix is small, we can compute them symbolically using the characteristic polynomial. However, this is often impossible for larger matrices, in which case we must use a numerical method.

Symbolic computations using the characteristic polynomial

An important tool for describing eigenvalues of square matrices is the characteristic polynomial: saying that λ is an eigenvalue of A is equivalent to stating that the system of linear equations $(A-\lambda I)v = 0$ (where I is the identity matrix) has a non-zero solution v (namely an eigenvector), and so it is equivalent to the determinant $det(A-\lambda I)$ being zero. The function $p(\lambda) = det(A-\lambda I)$ is a polynomial in λ since determinants are defined as sums of products. This is the characteristic polynomial of A: the eigenvalues of a matrix are the zeros of its characteristic polynomial.

It follows that we can compute all the eigenvalues of a matrix A by solving the equation $p_A(\lambda) = 0$. If A is an n-by-n matrix, then p_A has degree n and A can therefore have at most n eigenvalues. Conversely, the fundamental theorem of algebra says that this equation has exactly n roots (zeroes), counted with multiplicity. All real polynomials of odd degree have a real number as a root, so for odd n, every real matrix has at least one real eigenvalue. In the case of a real matrix, for even and odd n, the non-real eigenvalues come in conjugate pairs. An example of a matrix with no real eigenvalues is the 90-degree clockwise rotation:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

whose characteristic polynomial is $\lambda^2 + 1$ and so its eigenvalues are the pair of complex conjugates $i, -i$. The associated eigenvectors are also not real.

Eigen Value Theorem

In practice eigenvalues of large matrices are not computed using the characteristic polynomial. Computing the polynomial becomes expensive in itself, and exact (symbolic) roots of a high-degree polynomial can be difficult to compute and express (for example the Abel-Ruffini theorem implies that they cannot be expressed simply using nth roots. Effective numerical algorithms for approximating roots of polynomials exist, but small errors in the eigenvalues can lead to large errors in the eigenvectors. Therefore, general algorithms to find eigenvectors and eigenvalues, are iterative. The easiest method is the power method: we choose a random vector v and compute $Av, A^2v, A^3v, \ldots$ This sequence will after normalization almost always converge to an eigenvector corresponding to the dominant eigenvalue. This algorithm is easy, but not very useful by itself. However, popular methods such as the QR algorithm are based on it.

Analysis

Schrödinger's equation is a wave equation in terms of the wave function of physical variable in terms of expectation values and it is based upon the postulates of Quantum mechanics is time dependent equation which yields the time independent equation useful calculating the energy eigen values. It deals with ideas like uncertainty principle. For example, the energy Eigen value of the quantum harmonic oscillator are given by:

$$E_n = (n + 1/2)\bar{h}w.$$

$$w = 2\Pi(frequency)$$

$$\bar{h} = Planck'sconstant/2\Pi(6.6256 * 10^{-34} js)/2\Pi$$

$$n = Numberofsignalpresent.$$

Lowest energy of an oscillator is obtained by $n = 0$ is given by the equation $E_0 = 1/2(\hbar w)$. This equation is called Ground State Energy Equation. The Zero point energy is characteristic of the quantum mechanics and it is denoted by the equation $E_n = (2n+1)E_0$, where, $n = 0, 1, 2, 3, \ldots$

Whenever we make a measurement on a Quantum system, the results are dictated by the wave function at the time at which the measurement is made. It turns out that for each possible quantity we might want to measure (an observable) there is a set of special wave functions (known as eigen functions) which will always return the same value (an eigen value) for the observable. e.g.

EIGENFUNCTION	always returns
EIGENVALUE	
$psi_1(x,t)$	a_1
$psi_2(x,t)$	a_2
$psi_3(x,t)$	a_3
$psi_4(x,t)$	a_4
Etc...	etc...

Where (x,t) is standard notation to remind us that the Eigen functions $psi_n(x,t)$ are dependent upon position (x) and time (t).

Even if the wave function happens not to be one of these eigen functions, it is always possible to think of it as a unique superposition of two or more of the Eigen functions, e.g... $Psi(x,t) = c_1 * psi_1(x,t) + c_2 * psi_2(x,t) + c_3 * psi_3(x,t) + \ldots$ Where $c_1, c_2, \ldots$ are coefficients which define the composition of the state.

Consider a measurement of the position of a particle. Before the measurement is made the particle wave function is a superposition of several position Eigen functions, each corresponding to a different possible position for the particle, when the measurement is made the wave function collapses into one of these Eigen functions, with a probability determined by the composition of the original superposition. One particular position will be recorded by the measurement: the one corresponding to the Eigen function chosen by the particle.

If a further position measurement is made shortly afterwards the wave function will still be the same as when the first measurement was made (because nothing has happened to change it), and so the same position will be recorded. However, if a measurement of the momentum of the particle is now made, the particle wave function will change to one of the momentum Eigen functions (which are not the same as the position Eigen functions). Thus, if a still later measurement of the position is made, the particle will once again be in a superposition of possible position Eigen functions, so the position recorded by the measurement will once again come down to probability. What all this means is that one cannot know both the position and the momentum of a particle at the same time because when you measure one quantity you randomize the value of the other.

Notation: x=position, p=momentum

Action	wave function after action
Start	superposition of x and/or p Eigen functions
Measure x	x Eigen function = superposition of p Eigen functions
Measure x again	same x Eigen function
Measure p	p Eigen function = superposition of x Eigen functions
Measure x again	x Eigen function (not necessarily same one as before)

4 Analysis Results

In quantum mechanics, when you take a measurement of a system with state vector (wave function) $|\psi\rangle$ where the correspondingmeasurement operator $\hat{O}$ has eigenstates $|n>$ for $n = 1, 2, 3, \ldots$, and if you found one definite result value O_N then the state which the system had in this trial is consequently represented as $|N>$, the system may be said having been forced or "collapsed" into the state $|N\rangle$.The case of a continuous spectrum is more problematic, since the basis has uncountably many eigenvectors. These can be represented by a set of Delta functions. Since the delta function is in fact not a function, and moreover, doesn't belong to the Hilbert space of square-integrable functions, this can causes difficulties such as singularities and infinite values. In all practical cases, the resolution of any given measurement is finite, and therefore the continuous space may be divided into discrete segments.

Another solution is to approximate any lab experiments by a "box" potential (which bounds the volume in which the particle can be found, and thus ensures a countable spectrum).

Wavefunction collapse

Given any quantum state which is a superposition of eigenstates $|\psi>=c_1e^{-iE_1t}|1\rangle+c_2e^{-iE_2t}|2\rangle+c_3e^{-iE_3t}|3\rangle+\ldots$, if we measure, for example, the energy of the system and receive E_n (this result is chosen randomly according to probability given by

$$Pr(E_n)=\frac{|A_n|^2}{\sum_k|A_k|^2}$$

than the system's quantum state instantly becomes $|\psi\rangle=e^{-iE_2t}|2\rangle$ so any further measurement of energy will always yield E_2.

The process in which a quantum state instantly becomes one of the eigenstates of the operator corresponding to the measured observable (precisely which eigenstate is random, though the probabilities are determined by the square of the amplitude with which that eigenstate contributes to the overall state) is called "collapse", or "wavefunction collapse". The collapse process has no trace or corresponding mathematical description in the mathematical formulation of quantum mechanics. Moreover, the Schrodinger equation, which determines the evolution of the system in time, does not predict such a process, yet the process of collapse was demonstrated in many experiments (such as the double-slit experiment). The wavefunction collapse raises serious questions of determinism and locality, as demonstrated in the EPR paradox and later in GHZ entanglement.

There are two major approaches toward the "wavefunction collapse". This approach was supported by Niels Bohr and his Copenhagen interpretation which accepts the collapse as one of the elementary properties of nature (at least, for small enough systems). According to this, there is an inherent randomness embedded in nature, and physical observables exist only after they are measured (for example: as long as a particle's speed isn't measured it doesn't have any defined speed). This approach says that there is no collapse at all, and we only think there is. Those who support this approach usually offer another interpretation of quantum mechanics, which avoids the wavefunction collapse.

5 Conclusion

The system performs analysis of the various features extracted and comes out with the solution for optimizing the speech recognition rate by adopting the methods to extract the best features.

References

[1] K.J. Arrow, "Rational Choice Functions and Orderings," *Economica*, Vol. 26, pp. 121–127, 1959.

[2] A. Banerjee, "Fuzzy Choice Functions, Revealed Preference and Rationality," *Fuzzy Sets and Systems*, Vol. 70, pp. 31–43, 1995.

[3] R. Bělohlávek, *Fuzzy Relational Systems. Foundations and Principles*, Kluwer, 2002.

[4] J. Fodor, M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Kluwer, Dordrecht, 1994.

[5] I. Georgescu, "On the Axioms of Revealed Preference in Fuzzy Consumer Theory," *Journal of Systems Science and Systems Engineering*, Vol. 13, pp. 279–296, 2004.

[6] I. Georgescu, "Consistency Conditions in Fuzzy Consumers Theory," *Fundamenta Informaticae*, Vol. 61, pp. 223–245, 2004.

[7] I. Georgescu, "Revealed Preference, Congruence and Rationality: A Fuzzy Approach," *Fundamenta Informaticae*, Vol. 65, pp. 307–328, 2005.

[8] Roger A. Horn, Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, ISBN 0-521-30586-1, 1985.

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 294-300

[9] John B. Fraleigh, Raymond A. Beauregard, *Linear Algebra (3rd edition)*, Addison-Wesley Publishing Company, ISBN 0-201-83999-7 (international edition), 1995.

[10] Claude Cohen-Tannoudji, *Quantum Mechanics*, Wiley, ISBN 0-471-16432-1, 1977.

[11] Y. C. Eldar, A. V. Oppenheim, *Quantum Signal Processing*, Signal Processing Mag., vol. 19, pp. 12-32, Nov., 2002.

[12] L.R.Rabiner, B.H.Juang, *Fundamentals of speech recognition*, PTR Prentice Hall.

S. Sasikumar, S.Karthikeyan
P.S.N.A.College of Engineering and Technology
Department Of Electronics and Communication Engineering
E-mail: sonapoppy@yahoo.co.in, skarthik_82 @rediffmail.com

Collaborative Optimization in Dynamic Environments

Rodica Ioana Lung, Dan Dumitrescu

Abstract: A new evolutionary approach of dynamic environment optimization is proposed. A evolutionary population of candidate solutions and a swarm of particles collaborate in order to detect the moving global optimum of a problem. Mainly the EA population is used to exploit the search space while the swarm explores the space. The algorithm is tested on the moving peaks benchmark.

Keywords: evolutionary optimization, dynamic environments

1 Introduction

Enhancing evolutionary algorithms (EAs) with a tracking mechanism for moving optima is a realistic an necessary goal in view of the dynamic character of most real world applications. Examples of such applications are portofolio optimization, scheduling, vehicle routing, control problems. Because of their potential for adaptability, evolutionary algorithms are considered to be performant optimization tools for dynamic problems.

Another powerful nature inspired optimization method that has been widely used is the particle swarm optimization [7]. Inspired from the swarm theory it mimics the movement of a swarm in order to detect optima of a problem.

A change in the environment is considered to be a change on the fitness landscape. It is possible for an optimum to change its place and its value in time. The changes may occur randomly or they may follow a pattern. Considering that in critical situations there is no way to predict the kind of change will take place, an optimization tool should be able to adapt to random severe changes in the fitness landscape. In order to reduce the costs of the search, it would be benefic also to be able to track moving optima when they follow a certain trail.

A new approach to optimization in dynamic environments is proposed. It is a collaborative approach that combines the search abilities of an evolutionary algorithm and those of a particle swarm optimizer. The method aims at enhancing the qualities of these search techniques by bringing them to collaborate. Many such collaboration schemata can be devised. In order to illustrate the method potential in the Experimental section we focused on a particular model.

2 Evolutionary Optimization in Dynamic Environments

When applying evolutionary algorithms to dynamic optimization problems several issues may appear. After population convergence in a multimodal landscape it would be difficult for the optimizer to switch towards another optimum located in a different area of the search space. An diversity preserving mechanism is a minimal requirement to realize that.

It seems that one of the best method to ensure diversity within a dynamical environment is the use of multiple populations. Such methods are the Self-Organizing Scouts [3] or the Multinational GA [10].

Another way to adapt to changing environments is to use a memory tool [11, 5, 8]. However, such a memory tool is really useful only when optima return in previous positions. In a general situation it works like a supplementary diversity preserving mechanism.

3 PSO for dynamic environments

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart [7] in 1995. It is an efficient stochastic method of optimization inspired from particle swarm theory.

Hu and Eberhart [6] studied the behaviour of the swarm in different scenarios of adaptation to environment dynamics. Different detection methods and responses to the changes were studied and compared. A good method to track moving optima is to re-randomize a part of the swarm.

Blackwell and Bentley [1] imposed a diversificated structure in the swarm by charging particles and thus enlarging the area that can be 'covered' by the swarm.

Table 1: CESO Algorithm

```
begin
initialize EApopulation;
evaluate EApopulation;
initialize Swarm;
evaluate Swarm;
while not termination condition do
  begin
  if a change is detected then reinitialize swarm; evaluate EA population and Swarm;
  evolve EApopulation;
  update Swarm;
  evaluate EApopulation;
  evaluate Swarm;
  Enhance Collaborative mechanism - transmit information between populations
  end;
end.
```

Another way to deal with moving optima is to cause each particle to reset its record of its best position as the environment changes, to avoid making direction and velocity decisions on the basis of outdated information [4].

Branke and Blackwell proposed a combined technique that extends the single population PSO and Charged Particle Swarm Optimization (CPSO) methods by constructing interacting multi-swarms.

A Unified PSO scheme has been proposed by Parsopoulos and Vrahatis [9]. This scheme harnesses the local and global variant of PSO, combining their exploration and exploitation abilities.

4 Collaborative Evolutionary-Swarm Optimization

The observations in the previous sections indicate that EAs and PSO are merely complementary than competitive techniques. Therefore arises the idea that they may be used together in a search process .

We suggest a collaborative approach involving an evolutionary population an a swarm that evolve in parallel and exchange information. A new stochastic approach to dynamic optimization problems - called Collaborative Evolutionary-Swarm Optimization (CESO) - based on this idea is proposed.

The Swarm transmits information to the EA population regarding its position. At each iteration the worst individual in the EA population is replaced by *gbest* - the best particle in the swarm. This is a simple communication method that relates the swarm to the EA population. Other collaboration protocols can be designed.

Within a dynamical environment changes in the landscape are tested by re-evaluating the best individual in the EA population and testing its new fitness value against the old one. Everytime a change occurs the swarm is reinitialized in order to locate the new optimum.

Changes in the fitness landscape can vary in severity. If a small change occurs, the EA population can easily detect the new optimum. If the change is drastic, the Swarm indicates to the EA population the region of the new optimum.

CESO algorithm is outlined in Table 1. The EA population and the Swarm are randomly generated and then evaluated. An iteration of the algorithm is repeated until a termination condition is met. At the beginning of each iteration a test is performed to check if a change occurred in the environment during the last iteration. If a change did took place then the Swarm is regenerated and both populations are re-evaluated. Although this re-evaluation seems to be costly it actually prevents the EA population to evolve according to inadequate fitness values. In this way an iteration of the algorithm is practically saved.

The iteration continues by evolving the EA population under EA rules, i. e. applying selection and variation operators. The Particle Swarms are changed according to their velocity [7]. After the offsprings are evaluated, a transmission of information - generating collaboration - takes place between the EA population and the Swarm.

Remark 1. In order to emphasize the potential of the method, standard versions of both evolutionary algorithms and particle swarms are used. EA is not endowed with any supplementary diversity preserving mechanism (other then the in-built one of mutation) and the particle swarms velocity is computed using a constant *vmax*.

Table 2: Parameter setting for the moving peaks function for the three experiments

Parameter	**Experiment 1**	**Experiment 2**	**Experiment 3**
No of peaks	20	20	20
Dimensions	2	5	10
Search space	$[-100,100]^2$	$[-100,100]^5$	$[-100,100]^{10}$
Max peak height	10	100	100
Max peak width	1	1	1
Severity of change	0.5	1	20
Height severity	1	10	10
Width severity	1	10	10
λ	0	0	0
Base function	$f(x) = 0$	$f(x) = 0$	$f(x) = 0$
Frequency of change	30 generations	100 generations	300 generations
Peak function	cones	cones	cones

Table 3: Parameter setting for the algorithms

	Experiment 1		**Experiment 2**		**Experiment 3**	
Parameter	CESDO	SW	CESDO	SW	CESDO	SW
EA population size	50	-	100	-	200	-
Swarm size	50	100	100	200	200	400
No. of generations	10000		10000		90000	
No. of time spans	333		100		333	
$Vmax$	1	1	1	1	5	5
$c1, c2$	2,2					
Selection	tournament	-	tournament	-	tournament	-
Recombination	convex	-	convex	-	convex	-
Mutation	uniform	-	uniform	-	uniform	-

5 Experimental results

5.1 Experimental set-up

CESO is tested using the moving peaks benchmark.

The results are compared to a PSO algorithm that resets its *Pbest* every time a change occurs in the landscape. This a slight modification of the method proposed by Carlisle in [4].

CESO algorithm performs better than the simple PSO adapted to the dynamic environment. The reason is that when severe changes occur in the landscape it is difficult for the swarm to follow the optimum. In our method the EA population responds to the changes in the fitness landscape in a natural manner while the Swarm is regenerated and indicates to the EA population which is the new promising region. If the changes in the fitness landscape are minor, the EA population adapts easily. If the changes are major, the Swarm discovers the new optimum region.

Results of three experiments of varying difficulty are presented. The parameters of the function for the three tests are given in Table 2.

The first set up is a simple two-dimensional instance of the moving peaks with small severity of change. The second one is a 5-dimensional function and the third one is a 10 dimensional function with higher severity of change.

The parameters used to run the algorithms are presented in Table 3.

5.2 Results

Several performance metrics to compare the working of optimizers in dynamic landscape have been used in literature. One of them is the offline error, [2] defined as the average of all current errors, i.e. the average deviation of the currently best individual from the optimum. The current error represents the deviation of the best individual evaluated since the last change from the optimum.

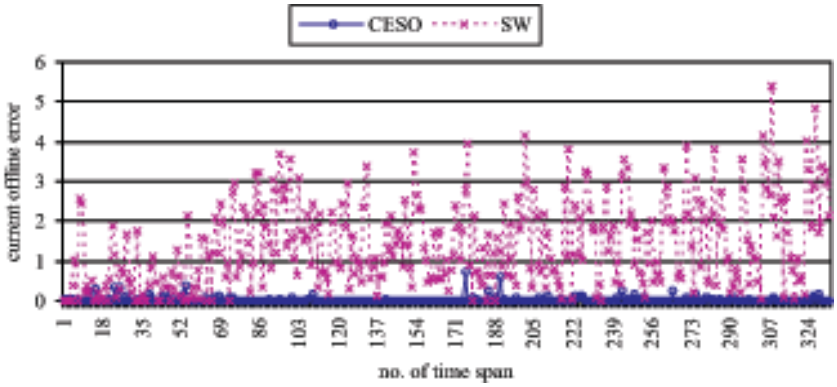


Figure 1: Current offline error for each time span - Experiment 1

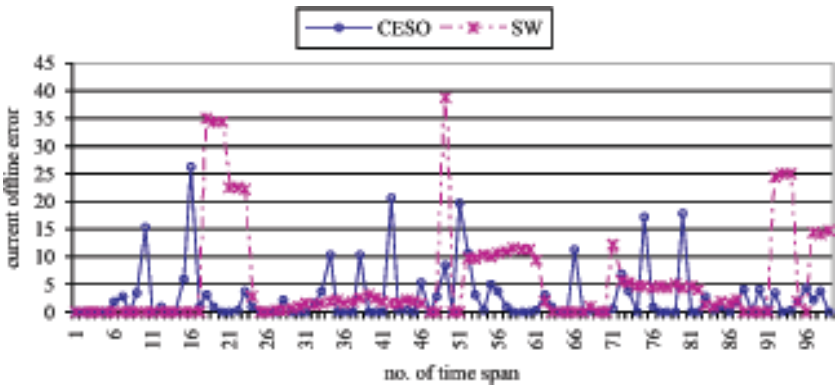


Figure 2: Current offline error for each time span - Experiment 2

Figures 1, 2 and 3 illustrate current offline errors at each time span for the three experiments. Numerical values for the offline errors and the corresponding standard deviation are presented in Tables 4 and 5.

We notice from the experimental results that the offline error for CESO is lower than the one for the simple PSO.

We notice that CESO performed better in all three experiments, showing an offline error, as well as a standard deviation of current offline errors lower than the other method.

The conclusion to be drawn is that the combined 'forces' of the two methods (EA and PSO) work better than just one of them (PSO).

6 Conclusions and further work

A new collaborative approach to dynamic optimization called Collaborative Evolutionary Swarm Optimization (CESO) is proposed. Proposed technique is based on the collaboration between an EA population and a Particle Swarm. The main purpose is to enhance the abilities of both methods by making them collaborative. Within CESO two populations, an evolutionary population and a particle swarm evolve in parallel. Each iteration they exchange information in order to benefit from their experiences. A simple model in which the swarm sends its *gbest* to the

Table 4: Averaged offline errors for Experiments 1 and 2

	Experiment 1		**Experiment 2**	
	CESO	SW	CESO	SW
Offline error	0.027949681	1.371622975	2.723038507	5.526716871
Standard deviation of offline errors	0.077368747	1.087486472	5.043218271	8.830340234

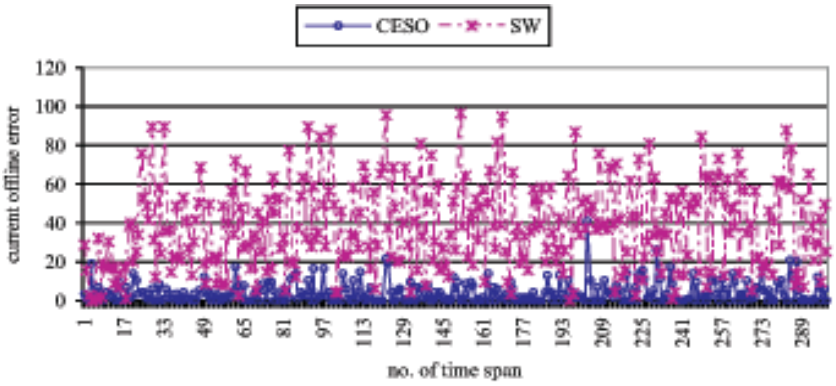


Figure 3: Current offline error for each time span - Experiment 3

Table 5: Averaged offline errors for Experiment 3

	Experiment 3	
	CESO	SW
Offline error	4.171086141	37.19516947
Standard deviation of offline errors	5.443470547	22.41085032

EA population is proposed.

CESO is tested on the moving peaks benchmark. Numerical experiments indicate that CESO outperforms a PSO enhanced with a simple adaptive mechanism for dynamic environments.

Further research will investigate new communication protocols and new collaborative models.

References

[1] T. M. Blackwell and P. J. Bentley. Dynamic search with charged swarms. In W. B. Langdon and al., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 19-26, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[2] Jurgen Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, 2001.

[3] Jurgen Branke, Thomas KauSSler, Christian Schmidt, and Hartmut Schmeck. A multi-population approach to dynamic optimization problems. In I.C. Parmee, editor, *Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM 2000)*, pages 299-308. Springer Verlag, 2000.

[4] Anthony Carlisle and Gerry Dozier. Adapting particle swarm optimization to dynamic environments. In *Proceedings of the International Conference on Artificial Intelligence 2000*, pages 429-434, 2000.

[5] David E. Goldberg and Robert E. Smith. Nonstationary function optimization using genetic algorithms with dominance and diploidy. In John J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates, Publishers, 1987.

[6] X. Hu and R. Eberhart. Adaptive particle swarm optimization: Detection and response to dynamicsystems. In David B. Fogel et al., editor, *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 1666-1670. IEEE Press, 2002.

[7] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In Proceedings of the *1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942-1948, Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995.

[8] Naoki Mori, Seiji Imanishi, Hajime Kita, and Yoshikazu Nishikawa. Adaptation to changing environments by means of the memory based thermodynamical genetic algorithm. In Thomas Back, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, pages 299Ű306, San Francisco, CA, 1997. Morgan Kaufmann.

[9] K.E. Parsopoulos and M.N. Vrahatis. UPSO: A unified particle swarm optimization scheme. In I*nt. Conf. Comput. Meth. Sci. Eng. (ICCMSE 2004)*, page 868-873, The Netherlands, 2004. VSP International Science Publishers.

[10] Rasmus K. Ursem. Multinational GAs: Multimodal optimization techniques in dynamic environments. In *Proceedings of the Second Genetic and Evolutionary Computation Conference (GECCO- 2000)*, volume 1, pages 19-26, Riviera Hotel, Las Vegas, USA, 2000. Morgan Kauffmann Publishers.

[11] Shengxiang Yang. Population-based incremental learning with memory scheme for changing environments. In Hans-Georg Beyer et al., editor, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 711-718, Washington DC, USA, 25-29 June 2005. ACM Press.

Rodica Ioana Lung, Dan Dumitrescu
Babeş-Bolyai University of Cluj Napoca
Department of Computer Science
Address: 1, Mihail Kogalniceanu, Cluj Napoca, Romania
E-mail: {srodica,ddumitr}@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 301-307

A New Key Exchange Protocol

Banshider Majhi, Y Santhosh Reddy, A.K. Turuk

Abstract: Diffie-Hellman protocol is the only strong way for key exchange in modern cryptography. It is widely using protocol even there are some inherent drawbacks in that. MQV is the efficient Diffie-Hellman protocol that use public key authentication. Ephemeral keys in the MQV protocol are insecure at the time of key exchange. Here we proposed one protocol which uses Diffie-Hellman protocol but session keys are not exposed directly. And for many attcks this protocol is secure than existing protocols that was proven in this paper. Security model for the key exchange protocols also discussed here.

Keywords: Key Exchange Protocols, MQV, Universal Composability.

1 Introduction

Diffie-Hellman key exchange protocol is most powerful one in the modern cryptology. While the basic protocol as originally proposed is believed to be secure against an eavesdropping-only attcker, the quest for an "authenticated Diffie-Hellman" protocol that resists active, man-in-the-middle, attacks has resulted in innumerable ad-hoc proposals, many of which have been broken or shown to suffer from serious weaknesses. Fortunately, with the development of rigorous security models for key exchange these days, we are in the position to judge the security of these protocols. Basic Diffie-Hellman key exchange protocol is discussed here.

1.1 Diffie-Hellman Key Exchange Protocol

This basic key exchange protocol is the starting point of modern cryptography. Diffie-Hellman protocol is based on the strong mathematical assumption called Discrete Logarithm Problem (DL Problem).

Discrete Logarithm Problem (DL Problem) Given input $\mathbb{F}_q$ (finite field), $g \in \mathbb{F}_q^*$ (generator of group) and $h \in_U \mathbb{F}_q^*$, it is computationally hard to find unique integer $a < q$ such that $h = g^a$.

The secrecy of the agreed shared key from the Diffie-Hellman key exchange protocol is exactly the problem of computing $g^{ab}(mod\, p)$ given g^a and g^b. This problem is called **computational Diffie-Hellman problem** (CDH problem). If the CDH problem is easy, then $g^{ab}(mod\, p)$ can be computed from the values p, g, g^a, g^b, which are transmitted as part of the protocol messages. The DL problem looks similar to taking ordinary logarithms in the reals. But unlike logarithms in the reals where we only need approximated "solutions", the DL problem is defined in a discrete domain where a solution must be exact.

Computational Diffie-Hellman Problem (CDH Problem) Given input $\mathbb{F}_q$ (finite field), $g \in \mathbb{F}_q^*$ (generator of group), g^a and g^b, it is computationally infeasible to find g^{ab}.

With a symmetric cryptosystem it is necessary to transfer a secret key to both communicating parties before secure communication can begin. An important advantage that public key cryptography is the achievement of a secure confidential channel. The first practical scheme to achieve this was proposed by Diffie and Hellman, known as the Diffie-Hellman exponential key exchange protocol[4].

Protocol 1: The Diffie-Hellman Key Exchange Protocol[4]
Input (p,g);p is a large prime, g is a generator element in $g \in \mathbb{F}_q^*$.
Output An element in $g \in \mathbb{F}_q^*$ shared between Alice and Bob.

a. Alice picks $a \in_U [1, p-1)$; computes $g_a \leftarrow g^a(mod\, p)$; sends g_a to Bob;

b. Bob picks $b \in_U [1, p-1)$; computes $g_b \leftarrow g^b(mod\, p)$; sends g_b to Alice;

c. Alice computes $k \leftarrow g_b^a(mod\, p)$;

d. Bob computes $k \leftarrow g_a^b (mod\, p)$;

Both Alice and Bob computes the shared key g^{ab}, third person can't compute g^{ab} from g_a and g_b because of CDH problem. This is first protocol which provides security against passive attacks but man-in-the-middle attack is the drawback. It is also insecure against active attacks.

1.2 Using Session Keys

In recent key exchange protocols session keys are using to make secure against longterm key attacks. These ephemeral exponents destroy by the principals after communicating with other. Principals are peers which want to share message. If these exponents are revealed, those are not useful for other sessions. MQV protocol is using session keys which are ephemeral. There are some attacks against MQV revealed last year. MQV protocol has been chosen by the NSA (National Security Agency) as a key exchange mechanism underlying "the next generation cryptograpgy to protect US government information". HMQV is the another recent key exchange protocol which adapted properties from MQV protocol but with slight change. In the original paper of HMQV author was stated that it is secure than MQV protocol. Here we introducing one new key exchange protocol which is secure against some attacks than any existing protocol. We used session keys to make protocol secure against longterm key attacks.

1.3 Organization of paper

The paper is organized in the following sections: Section 2 provides the description of the model of secure key-exchange protocols. Section 3 introduces new key exchange protocol. Section 4 provides security against different attacks. Section 5 discusses MQV and HMQV protocols. Section 6 gives conclusion and further working direction.

2 Security Analysis

A key-exchange (KE) protocol is run in a network of interconnected parties where each party can activated to run an instance of the protocol called session. Within a session a party can be activated to initiate the session or to respond to an incoming message. As a result of these activations, and according to the specification of the protocol, the party creates and maintains a session state, generates outgoing messages, and eventually completes the session by outputting a session key and erasing the session state. A session may also be aborted without generating the session key. A KE session is assosiated with its holder or owner, a peer, and a session identifier [3]. We are taking two assumptions here **(i)** the activation of a session at a party always specifies the name of the intended peer; and **(ii)** the session identifier is a quadruple $(\hat{A}, \hat{B}, Out, In)$ where $\hat{A}$ is the identity of the holder of the session, $\hat{B}$ is the peer, *Out* the outgoing messages in the session, and *In* the incoming messages. In particular, in the case of MQV and HMQV this results in an identifier of the form $(\hat{A}, \hat{B}, X, Y)$ where X is the outgoing DH value and Y is the incoming DH value to the session. The peer that sends the first message in a session is called initiator and other the responder. The session $(\hat{A}, \hat{B}, X, Y)$ is matching to the session $(\hat{B}, \hat{A}, Y, X)$. As we know matching sessions play a fundamental role in the definition of security.

Analysis of Attack

The attacker is modeled to capture realistic attack capabilities in open networks, including the control of communication links and the access to some of the secret information used or generated in the protocol. The basic principle is that since security breaches happen in practice, a well-designed protocol needs to confine the damage of such breaches to a minimum. In perticular, the leakage of session-specific ephemeral secret information should have no adverse effect on the security of any other non-matching sessions[2].
The attacker $\mathcal{M}$ is an active "man-in-the-middle" adversary with full control of the communication links between parties. $\mathcal{M}$ can intercept and modify messages sent over links, it can delay or prevent their delivery, inject its own messages, interleave messages from different sessions, etc. $\mathcal{M}$ also schedules all session activations and session-message delivery. The attacker is allowed acess to secret information via session exposure attacks of three types: *state reveal queries, session-key queries*, and *party corruption*.

A state reveal query is directed at a single session while still incomplete and its result is that the attacker learns the session state for perticular session. A session-key query can be performed against an individual session after completion and the result is that the attacker learns the corresponding session-key. A party corruption means that the attacker learns all information in the memory of that party (including longterm key) . Indeed, the knowledge of the private key allows the attacker to impersonate the party. Sessions against which any one of the above attacks is performed are called exposed. In addition, a session is also called exposed if the matching session has been exposed.

The security of session keys generated in unexposed sessions is captured via the inability of the attacker $\mathcal{M}$ to distinguish the session key of a test session, chosen by $\mathcal{M}$ among all complete sessions in the protocol, from a random value. When $\mathcal{M}$ choses the test session it is provided with a value v which is chosen as follows: a random bit b is tossed, if $b = 0$ then v is the real value of the session key, otherwise v is a random value chosen under the same distribution of session-keys produced by the protocol but independent of the value of the real session key. After receiving v the attacker may continue with the regular actions against the protocol: at the end of its run $\mathcal{M}$ outputs a bit b'. The attacker suceeds in its distinguishing attack if (1) the test session is not exposed, and (2) the probability that $b = b'$ is significantly larger than 1/2.

Security of Key Exchange A polynomial-time attacker with the above capabilities is called a KE-attacker. A key-exchange protocol π is called secure if for all KE-attackers $\mathcal{M}$ running against π it holds:

a. If two uncorrupted parties complete matching sessions in a run of protocol π under attcker $\mathcal{M}$ then, except for a negligible probability, the session key output in these sessions is the same.

b. $\mathcal{M}$ suceeds with probability not more than 1/2 plus a negligible fraction.

An important property of key-exchange protocols not captured in the above definition is perfect forward secrecy (PFS)[8], namely the assurance that once a session key is erased from its holders memory then the key cannot be learned by the attcaker even if the parties are subsequently currupted.

3 New Key Exchange Protocol

In conventional Diffie-Hellman protocol ephemeral exponents are directly sending to the other party, because of this "man-in-the-middle" attacks are easy eventhough authentication is there. And there may chance to deniable authentication because of the no involvement of public key cryptography in basic key exchange. Here we presenting the protocol which exposes another view of the Diffie-Hellman protocol. Up to now primary communication is based on only the ephemeral values. $\hat{A}$ picks one value x in the group and send $g^x = X$ to $\hat{B}$. $\hat{B}$ also picks one value y and send $g^y = Y$ to $\hat{A}$. After that these session values are using in the protocol and finally gets one session key.

Here we introduces another way of Diffie-Hellman protocol. $\hat{A}$ randomly generate one group element x and send $B^{ax} = I$ to $\hat{B}$ (where a is private key of $\hat{A}$ and A is public key of $\hat{A}$). Similarly $\hat{B}$ randomly generate one group element y and send $A^{by} = J$ to $\hat{A}$ (where b is private key of $\hat{B}$ and B is public key of $\hat{B}$). The basic key exchange is shown in *Figure 1*.

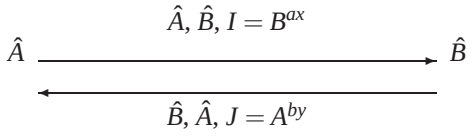


Figure 1: The variation of Diffie-Hellman Key Exchange

3.1 Protocol

The communication in this protocol is inherits the key exchange in Figure 1. Protocol is described in Figure 2.

Session Key Computation in New Protocol

a. $\hat{A}$ and $\hat{B}$ run Key Exchange shown Figure 1

b. $\hat{A}$ computes $d = \hat{H}(J^x, B)$, $e = \hat{H}(J^x, A)$

c. $\hat{B}$ computes $d = \hat{H}(I^y, B)$, $e = \hat{H}(I^y, A)$

d. $\hat{A}$ computes $t_{\hat{A}} = (JB^{ae})^{x+d}$, $\hat{B}$ computes $t_{\hat{B}} = (IA^{bd})^{y+e}$

e. Both $\hat{A}$ and $\hat{B}$ set session key $K = H(t_{\hat{A}}) = H(t_{\hat{B}})$

Figure 2: Computation of session key K
(A and B are public keys of $\hat{A}$ and $\hat{B}$ respectively)

In above protocol two hash functions $\hat{H}$ and H are used to make secure against Key Compromise Impersonation (KCI) attacks. Section 4 discusses the security analysis against various attacks.

Finally both $\hat{A}$ and $\hat{B}$ computes the same session key (it is proven with simple observation). Because of using public keys of $\hat{A}$ and $\hat{B}$ in the computation of d and e, this protocol succeeded in deniable authentication. It is not sending ephemeral key exponents solely (as in all other modern key exchange protocols), because of this even the attacker know the private key of parties he can't find the session key. Main security advantage of this protocol is using Multiple Exponents (including private keys of the parties) in the computation of session key and both temporary variables d and e.

Even though no security problems in this protocol there may be chance to insecure against Random Session Key (RSK) attacks. It is because of the multiple exponents of generator g, which will not distribute session keys uniformly in the Field. By the selection of perfect generator we can easily overcome this problem, it needs additional screening. Even one of the parties corrupted by $\mathcal{M}$ this protocol tends security (because of computationally hard problem of finding N root). This is not in the scope of paper so we are avoiding to discuss. Various attacks and security justification is discussed in next section. After that two recent key exchange protocol MQV and HMQV are discussed in Section 5.

4 Security of Protocol

This is most suitably designed against various attacks. In Diffie-Hellman Key Exchange protocols this is the novel direction. Even though todays key exchange protocols are very secure against different attacks, there might be some property sacrificed. But in this New Protocol security compromise will not appear. Even it is similar in computing d and e values as in HMQV protocol[6], it is proving Perfect Forward Secrecy (PFS). There is no insecurity of computing d and e as constants like MQV protocol[7]. From here definitions of different attacks and the simple security justification presented (Detailed security proofs are not presenting here). Some attacks and validations which are Key Compromise Impersonation (KCI) attack, Unknown Key Share (UKS) attack[5], Perfect Forward Secrecy (PFS), basic impersonation attacks, longterm public key validation [2] and disclosure of Diffie-Hellman exponents are discussed here.

Key Compromise Impersonation (KCI) attack In case $\mathcal{M}$ has learned the private key a of a party $\hat{A}$. $\mathcal{M}$ can now impersonate $\hat{A}$ to any other party, this is called Key Compromise Impersonation attack. But even in this situation it is disirable to prevent $\mathcal{M}$ from impersonating other parties to $\hat{A}$. this property, referred to as resistance to KCI attcks.

This protocol is *resistance to KCI attacks* because even though private key exposed session key x was not directly sending and hashing of $t_{\hat{A}}$ and $t_{\hat{B}}$ values. This is not the perfect attack for todays key exchange protocols. Assume that attacker also knows the hash function H and ephemeral key x impersonation will not occure because of another hash function $\hat{H}$.

"Man-in-the-middle" attack is not discussed in this Section because all other attacks discussed here are recent attcks than that.

Perfect Forward Secrecy (PFS) A key-exchange protocol is said to have PFS property if the leakage of the long-term key of a party does not compromise the security of session keys established by the party and erased from memory before the leakage occured.

Even the party was corrupted by $\mathcal{M}$ the session keys will not be exposed. The powerful advantage of this protocol is it provides PFS with 2-message protocol, but no other 2-message protocol provides PFS. To make confident we can use 3-message protocol version of this. But is is not required in practice.

Unknown Key Share (UKS) attack An unknown key share attack is perticular form of impersonation attack in which attacker $\mathcal{M}$ interferes with session establishment between two honest parties $\hat{A}$ and $\hat{B}$ such that at the end of the attack both parties compute the same session key K, yet while $\hat{A}$ is convinced that the key is shared with $\hat{B}$, $\hat{B}$ believes that the peer to the session has been $\mathcal{M}$.

This attack , first discovered by Diffie, van Oorschot and Wiener [8] , has become a basic attack under which key exchange protocols are validated. This attack will occure when attacker $\mathcal{M}$ register the public key which is using by the target party, but by the simple restriction on the selecting of public key will make secure aginst this attack. For detailed discussion about UKS attack see[5]. In this protocol UKS attack is impossible because of impersonation will not occure as explained for previous attacks. since attacker forces the same session key K to be computed in two *non-matching* sessions, $(\hat{A},\hat{B},X,Y)$ and $(B,\mathcal{M},Y,X)$, then the attacker can use former as the test session and win the game.

4.1 Long-term public key validation

Advantage of this protocol is no need to verify the identity of the public key. Generally this verification will done by *Certification Authority (CA)* at the time of the key registration. Yet, since not all *CA*'s will be configured to do such tests, it is better to minimize these extra requirements from *CA*s. In this new protocol a dishonest party can choose an element of order other than q as its public key and unable to cause harm any honest party.

4.2 Disclosure of Diffie-Hellman exponents

An important security requirement from Diffie-Hellman protocols is that the disclosure of the exponent of an ephemeral DH value should not compromise the security of the session key. This protocol can withstand the leakage of ephemeral DH exponents. This is because of we are not using x as single exponent and other one is CDH problem.

5 MQV and HMQV Protocols

The MQV protocol [1] of Law, Menezes, Qu, Solinas and Vanstone is possibly the most efficient of all known authenticated Diffie-Hellman protocols that use public-key authentication. The HMQV protocol is the extention of MQV protocol with using additional hash function. MQV protocol has been widely standardized, and has been chosen by the *NSA* as the key exchange mechanism underlying *"the next generation cryptography to protect US governament information"* [7].

Unfortunately MQV protocol was analyzed by the Canetti-Krawczyk model[3] of key exchange and proven insecure against some attacks mentioned in Section 4. The small variation of MQV protocol which overcomes those insecurities in MQV is HMQV (for details see[6]). These protocols are shown in Figure 4. Even though HMQV is better than MQV protocol in many ways it will not purely provides Perfect Forward Security (PFS). There is an inherent weakness in conventional Diffie-Hellman protocol is that 2-message protocols are always not proven PFS. But in our new protocol because of multiple exponents this problem was partially rectified. But for better performance we can use 3-message version of our protocol.

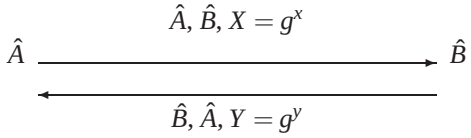


Figure 3: The basic Diffie-Hellman Key Exchange

Session Key Computation in MQV and HMQV protocols

a. $\hat{A}$ and $\hat{B}$ run basic Key Exchange (Figure 3)

b. $\hat{A}$ computes $t_{\hat{A}} = (YB^e)^{x+da}$, $\hat{B}$ computes $t_{\hat{B}} = (XA^d)^{y+eb}$

c. Both $\hat{A}$ and $\hat{B}$ set session key $K = H(t_{\hat{A}}) = H(t_{\hat{B}})$

MQV: $d = \bar{X} \equiv 2^l + (X mod 2^l)$ $e = \bar{Y} \equiv 2^l + (Y mod 2^l)$ $l = |q|/2$.
HMQV: $d = \bar{H}(X, \hat{B})$ $e = \bar{H}(Y, \hat{A})$ $l = |q|/2$.

Figure 4: MQV and HMQV protocols
(A and B are public keys of $\hat{A}$ and $\hat{B}$ respectively)

These two are todays most powerful key exchange protocols even though there are some drawbacks. The HMQV protocol is not having serious security loopholes but we want to explore multiple exponents view of Diffie-Hellman key exchange so we carefully designed new protocol. Finally our protocol provides all security goals including PFS.

We did not discussed much about security proofs of our new protocol. Upto now we did not found any loopholes in this new protocol except PFS of 2-message protocols is inherently weak.

6 Conclusion

In this paper we introduced new key exchange protocol which is better in the performance compared with most todays keyexchange protocols. Here deep discussion about security proofs of our protocol was omitted. These security proofs will be our future work.

Even though this new protocol provides security the computational complexity and efficiency are not sacrificed. We are believing this protocol will open new directions in key exchange protocols. In future game of attack against this protocol should be developed and universal composability will be checked.

References

[1] S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman Key Agreement Protocols", *Proceedings of SAC '99*, Lecture Notes in Computer science, 1556 (1999).

[2] S. Blake-Wilson, D. Johnson and A. Menezes, "Key exchange protocols and their security analysis", *Sixth IMA International Conference on Cryptography and Coding*, 1997.

[3] R. Canetti, "Universally Composable Security: A New paradism for Cryptographic Protocols", *42nd FOCS*, 2001.

[4] W. Diffie and M. Hellman, "New directions in Cryptography", IEEE Trans. Info. Theor. 22, 6(Nov 1976), pp. 644–654.

[5] W. Diffie, P. van Oorschot and M. Wiener, "Authentication and Authenticated key exchanges", *Designs, Codes, and Cryptography*, 2, 1992, pp. 107–125.

[6] H. Krawczyk, "HMQV: A High-Performance Secure Diffie-Hellman Protocol", In `http://eprint.iacr.org/2005/176`, December 2005.

[7] L. Law, A. Menezes, M. Qu, J.Solinas and S. vanstone, "An efficient Protocol for Authenticated Key Agreement", *Designs, Codes, and Cryptography*, 28, 119–134, 2003.

[8] A. Menezes, P. Van Oorschot and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[9] Wenbo Mao, "Modern Cryptography Theory and Practice", Pearson Education.

Banshider Majhi Majhi, Y Santhosh Reddy, A.K. Turuk
Department of CSEA
NIT Rourkela
India, 769008
E-mail: ysantosh@rediffmail.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 308-312

On the Use of Genetic Algorithms in Molecular Modeling

Annamaria Mesaros

Abstract: Molecular modeling is used in different research areas, nowadays combining chemistry, physics and engineering. In generating molecular data, quantum mechanics stands for the mathematical formalism in determining the minimum energy conformation. For empirically methods to solve the problems of conformational search, different optimization techniques can be used. The genetic algorithms are powerful engineering tools that provide parallel processing of information. Genetic algorithms can be successfully used in searching a minimum energy conformation of a polyatomic molecule, as long as the potential energy of the molecule is properly expressed for such a solving manner. The algorithm reduces to searching for an optimal minimum on a hypersurface depending on the variables of the studied molecule.

1 Introduction

Molecular modeling is a general term that covers a wide range of molecular graphics and computational chemistry techniques used to build, display, manipulate, simulate, and analyze molecular structures and to calculate properties of these structures. Molecular modeling is used in several different research areas in physics and chemistry. Computational chemistry/molecular modeling is the science of representing molecular structures numerically and simulating their behavior with the equations of quantum and classical physics. Computational chemistry programs allow scientists to generate and present molecular data including geometries, energies, electronic properties, and spectroscopic properties.

2 Quantum mechanics

Quantum mechanics is one of the oldest mathematical formalisms of theoretical chemistry. It uses the physical constants such as the velocity of light, values for the masses and charges of nuclear particles and differential equations to directly calculate molecular properties and geometries. This formalism is referred to as *ab initio* (first principle) quantum mechanics. The equation from which molecular properties can be derived is the Schrödinger equation:

$$H\Psi = E\Psi \tag{1}$$

where E is energy of the system relative to one in which all atomic particles are separated to infinite distances, Ψ is the wavefunction which defines the cartesian and spin coordinates of the atomic particles and H is the Hamiltonian operator which includes terms for potential and kinetic energy. The Schrödinger equation can be solved only for very small molecules such as hydrogen and helium, while for polyatomic systems approximations must be introduced.

The first approximation assumes that nuclei are much heavier than electrons and move much more slowly, so that molecular systems can be viewed as electrons moving in a field of fixed nuclei. This is the Born-Oppenheimer approximation. Solutions to the Schrödinger equation using this assumption lead to values of effective electronic energy, which are dependent on relative nuclear coordinates. As the nuclei are moved to new coordinates and molecular energies are re-calculated, a quantitative description of molecular energy is derived. This description relates energy to geometry and is referred to as the potential energy surface for the molecule. The lowest point on this surface with respect to energy is the ground state energy and its associated geometry for the molecule.

A second approximation, Hartree-Fock assumes that the energy of a set of molecular orbitals can be derived from the basis set functions, which are used to define each orbital and a set of adjustable coefficients, which are used to minimize the energy of the system. The energy calculation becomes solving a set of (NXN) matrices to obtain optimal values for the orbital coefficients. Since this calculation requires a value for the coefficients in order to solve the equations, an iterative process is used in which an initial guess for the value of the coefficients is progressively refined until it provides consistent values.

3 Molecular mechanics

Molecular mechanics attempts to reproduce molecular geometries, energies and other features by adjusting bond lengths, bond angles and torsion angles to equilibrium values that are dependent on the hybridization of an atom and its bonding scheme. The method relies on the laws of classical Newtonian physics and experimentally derived parameters to calculate geometry as a function of energy. The general form of the force field equation is:

$$E_{pot} = \sum E_{bnd} + \sum E_{ang} + \sum E_{tor} + \sum E_{nb} + \sum E_{el} \tag{2}$$

E_{pot} is the total energy, which is defined as the difference in energy between a real molecule and an ideal molecule. E_{bnd} is the energy resulting from deforming a bond length from its natural value; E_{ang} is the energy resulting from deforming a bond angle from its natural value. E_{tor} is the energy that results from deforming the torsion or dihedral angle. E_{nb} is the energy arising from non-bonded interactions and E_{el} is the electrostatic energy, arising from coulombic forces.

The manner in which the terms from eq. 2 are utilized is referred to as the functional form of the force field. The force constants and equilibrium values used to express these forces are atomic parameters that are experimentally derived using *ab initio* calculations on a given class of molecules. The energy of the atoms in a molecule is calculated and minimized using a variety of directional derivative techniques.

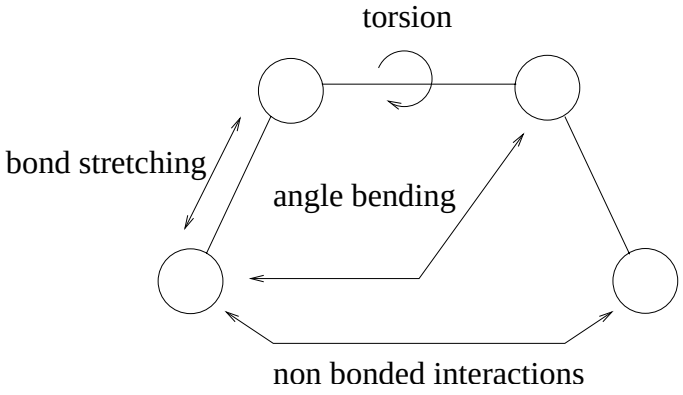


Figure 1: Elements of the force field equation in a molecule

Different methods are used to calculate the energy of a compound in a specific 3D orientation and to optimize the geometry as a function of energy (adjust the coordinates of each of the atoms and recompute the energy of the molecule until a minimum energy is obtained). Coupled with other numerical techniques, they also can be used to simulate the time-dependent behavior of molecules (molecular dynamics) and explore their conformational flexibility (conformational search).

4 Genetic algorithms

A genetic algorithm (GA) is a programming technique that mimics biological evolution as a problem-solving strategy. Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called fitness function that allows each candidate to be quantitatively evaluated. The GA evaluates each candidate according to the fitness function. Some candidates are kept and allowed to reproduce. Multiple copies are made of them, with random changes introduced during the copying process. These offspring then go on to the next generation, forming a new pool of candidate solutions, and are further evaluated. The process repeats, the expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be discovered.

There are many different techniques that a genetic algorithm can use to select the individuals to be copied over into the next generation. The most common methods are elitist selection, fitness-proportionate selection, scaling selection, generational selection, and rank selection. Once selection has chosen fit individuals, they must be randomly altered.

One of the qualities of genetic algorithms is that GAs know nothing about the problems they are deployed to solve. Instead of using previously known domain-specific information to guide each step and making changes with a specific eye towards improvement, as human designers do, they are "blind watchmakers"; they make random changes to their candidate solutions and then use the fitness function to determine whether those changes produce an improvement.

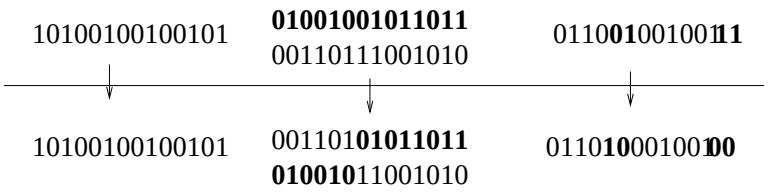


Figure 2: The three methods used by GA to obtain a new generation. Replication - the chromosome is not altered. Crossover - combines the characteristics of two chromosomes. Mutation - Each bit has a certain probability to change from 0 to 1 and vice-versa

5 Molecular conformation search using genetic algorithms

The potential energy cost for changing the bond length in a molecule has a minimum at the equilibrium value. We take as example a simple triatomic molecule, such as CO_2, confined to move along a straight line. The molecule consists of three atoms, A-B-C, as presented in figure 3.

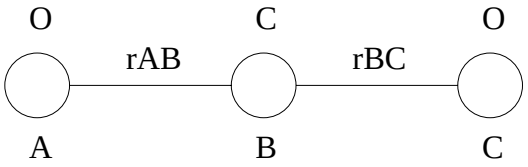


Figure 3: The simplified structure of a CO_2 molecule

Each of the two bonds can be described by a Morse interaction potential as in equations 3 and 4.

$$V_{AB} = d_{AB}(e^{-2a_{AB}(r-b_{AB})} - 2e^{-a_{AB}(r-b_{AB})}) \tag{3}$$

$$V_{BC} = d_{BC}(e^{-2a_{BC}(r-b_{BC})} - 2e^{-a_{BC}(r-b_{BC})}) \tag{4}$$

The parameters in these equations are d_{AB}, d_{BC} - the potential well depth in eV, a_{AB},a_{BC} - the decay length of potential in 1/Å, and b_{AB}, b_{BC} - the equilibrium bond length in Å. The total potential energy of the molecule is:

$$V(r_{AB}, r_{BC}) = V_{AB}(r_{AB}) + V_{BC}(r_{BC}) \tag{5}$$

By evaluating this function for a CO_2 molecule, we obtain the energy contour plot and energy surface plot in figure 4.

This is rather a simple surface for using genetic algorithms in solving for a minimum energy state, but it will be used as an example for implementing a GA in molecular modeling. The fitness function for the GA formulation problem is the potential function of the molecule, with two independent variables: the interatomic distances within the molecule. The structure of the CO_2 molecule is linear, thus we do not introduce any variable related to bending. For a double vector type population of size 20 and a uniform creation function, the GA parameter choices were: rank scaling function, gaussian mutation, two-point crossover. Figure 4 presents the fitness value for one generation of individuals and a fragment of the genealogy illustrating the combinations of genes between individuals in order to obtain the new generation.

After a number of 100 generations, the values returned for the two variables as determining the minimum potential energy are 1.16231 and 1.16209; these are the equilibrium bond lengths between atoms A-B and B-C, the C-O bond length in Å. For iterations with different GA options, the final point of the algorithm is slightly different. For bit string population type, the algorithm generated a large number of stall generations - no improvement in the objective function for a sequence of consecutive generations.

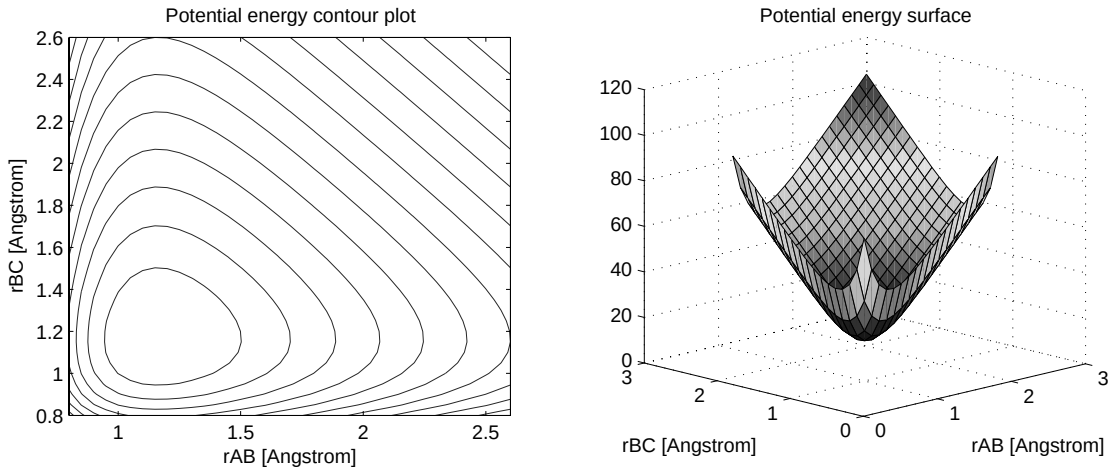


Figure 4: Contour and surface energy plot of a CO_2 molecule as a function of the inter-atomic distances

6 Conclusions

The results of the simulations show that it is very important to express the problem to be solved in a convenient form and with certain coding of its parameters, in order to obtain the best result using genetic algorithm. The form of the population and the evolutionist strategy counts for the final result. For polyatomic molecules and a more complex force field, a potential hypersurface must be expressed as a function of the n variables with respect to whom the function must be minimized. If the surface is very complex in terms of local minima, a genetic algorithm is much more likely to give an optimal solution, than the classical search algorithms.

If we think about computational complexity, the genetic algorithms provide parallelism and stochastic search in random areas of the hypersurface, unlike classical search that follows a path on the search surface. Genetic algorithms can be successfully used in searching a minimum energy conformation of a polyatomic molecule, as long as the potential energy of the molecule is properly expressed for such a solving manner. The algorithm reduces to searching for an optimal minimum on a hypersurface depending on the variables of the studied molecule.

References

[1] Dammkoehler, R.A, Karasek, S.F, Shands, E.F.B, Marshall, G.R., “Constrained Search of Conformational Hypersurface,” *J. Comput-Aided Mol. Design*, 3, 3, 1989

[2] Forrest, S., ”Genetic Algorithms: Principles of Natural Selection Applied to Computation” *Science*,, 261, 872-878, 1993

[3] Forrest, SMundim, K.C, Ellis, D.E., ”Stochastic classical molecular dynamics coupled to functional density theory: applications to large molecular systems,” *Braz. J. Phys.*,, Mar. 1999, vol.29, no.1, p.199-214. ISSN 0103-9733

[4] Kellö, V., Lawley, K.P, Diercksen, G.H.F, ”An ab-initio investigation of the dipole moment of the CO2...CO complex” *Chem. Phys. Letters*,, 319, pages 231-237, 2000

[5] Labanowski, J.K., ”Molecular Modeling” *computational chemistry list*,, http://www.ccl.net/

[6] Richon, A.B., ”An Introduction to Molecular Modeling” *Mathematech*,, 1, 83, 1994.

Annamaria Mesaros
Technical University of Cluj-Napoca
Communications Department
Address: G. Baritiu 26-28 RO-400027, Cluj-Napoca, Romania
E-mail: annamaria.mesaros@com.utcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 313-319

Intelligent Urban Traffic Signalling Infrastructure with Optimised Intrinsic Safety

Marius Minea, Florin Codruţ Nemţanu

Abstract: It is a known fact that the road transportation is one of the most affected with traffic incidents, casualties, traffic jams, and general congestion, due to the large number of vehicles, both private and/or belonging to state agencies or companies. In Romania, the number of vehicles increased tremendously in the last years, and the road infrastructure is oversaturated from its capacity point of view and unsafe, from its signalling point of view. The present paper tries to present some issues concerning the increase of the signalling infrastructure intrinsic safety. Some new methods to improve the reliability, both by new technologies for data acquisition and by remote equipment monitoring techniques are presented. The new conditions in the Information Society, implemented in all developed countries thru Intelligent Transport Systems (ITS) are also mentioned. The paper also shows some results from one of our research projects.

1 Introduction - International and National Situation

In the last years of the eighth decade, last millennium, the traffic density in the great cities (metropolitan areas) has reached impressive values. Among all transport modes, the road transportation has some determinant characteristics: the large number of single occupant vehicles, the missing traffic large area regulation systems, and all these factors lead to a large number of incidents, environmental pollution, fuel consumption and stress. Along with these issues, the presence of a large number of non-professional vehicle drivers lead also to a high incident rate, frequently producing human casualties.

Several developed countries like U.S.A., Australia, Japan, and in Europe: Great Britain, France, Germany, Belgium, Austria and others, have started, several years ago, national and international ITS infusion programmes, to implement IT, telecommunications and telematics technologies, in order to improve the traffic conditions, and its safety. Shortly after, these programmes were undertaken at governmental (political) and non-governmental levels, defining what today is known as “Intelligent Transport Systems” – a variety of specific applications that use the above technologies to provide information and support to all road users/owners, travellers, vehicle drivers, administrators etc.

Among the most important projects, at international level, developed in this field of activities, there are: “DRIVE” – Safety Infrastructure for the European Road Traffic, “PROMETHEUS” – the Program for a European Traffic with Highest Efficiency and Unprecedented Safety, or in USA, some programs for Intelligent Highways Infrastructure. Almost in the same period, in Japan there were developed several other programs and systems, like VICS[1], UTMS[2], ARTS[3] and SSVS[4].

Urban road networks serve a significant part of traffic demand. Because of the high demand many urban road facilities are frequently oversaturated and respectively congested. Through congestion the capacity of the road infrastructure is in fact reduced and particularly during rush hours when the maximum capacity is most urgently needed the performance deteriorates considerably. The principle of adaptive control was first used by Miller (1963) when he proposed a strategy that is based on an online traffic model. The model calculates time wins and losses and combines these criteria for the different stages in a performance index to be optimised. In sequence a series of adaptive methods were developed. A prominent example of the so called first generation of traffic adaptive strategies (which were not really adaptive) is PLIDENT (Holroyd et al., 1971). In the late 1960s PLIDENT was implemented in Glasgow and failed. Field trials with other first generation strategies in Canada (Corporation of Metropolitan Toronto, 1976) as well as in the frame of the UTCS programme in the United States (MacGowan et al., 1980) also failed due to inaccurate demand forecasts for longer time periods, slow reactions and capacity losses caused by transition programmes. Hunt et al. (1981) analysed the shortcomings of the 1st generation strategies and overcame the problems by a more advanced 2nd generation strategy. One of the first developments in the adaptive

[1] Vehicle Information and Communication System
[2] Urban Traffic Management System
[3] Advanced Road Traffic System
[4] Super Smart Vehicle System

control technology was SCOOT[5] concept. It minimizes delay by the smooth adaptation of split, cycle time and offset. In contrast to general believing only the offset is optimised on the basis of delay modelling whereas split and cycle times are adapted according to a saturation criterion. With successful trials of SCOOT in different networks resulting in savings of about 12% delay the break-through of adaptive methods succeeded. SCOOT is at present the most established control method with over 170 implementations all over the world. In the early 1980s a number of advanced so-called 3rd generation strategies have been developed, e.g. OPAC (Gartner, 1982) or PRODYN (Henry et al., 1983). These strategies are operating acyclic, i.e. they do not consider explicitly cycles or offsets. Given predefined stage schemes optimal switching times over a horizon are calculated. The optimization is based on delay criteria determined by simple but fast running traffic models. For the global optimization of the performance function OPAC employs complete enumeration whereas PRODYN uses dynamic programming. Due to the exponentially increasing complexity of the solution for more than one intersection real time implementations are only feasible for isolated intersections.

In Europe, the UTMC[6] systems are implemented for the first time in several cities, among which London, along with another intelligent system, the Congestion Fare Collection system, a measure to reduce traffic congestions in central London. The last is provided with CCTV and intelligent plate registration numbers recognition algorithms, recording and monitoring all vehicles that cross the system boundaries. Specific UTMC systems, such as SCOOT, SCATS[7] and UTOPIA[8], are now wide spread around the world and already considered "mature" UTMC systems.

An UTMC system, component of Intelligent Transport Systems, defines a generalised architecture for emissions and vehicle priority in term of:

- Monitoring/location of Public Transport Vehicles (PTVs), or other type of vehicles;
- Road Network Model;
- Traffic Management Strategy Selection;
- Strategy Implementation; and
- Common/historic databases.

Adaptive signal control has recently emerged as a viable system control strategy within the United States, Europe and Australia. Long used in Europe and Australia, adaptive control consists of the real-time adjustment of coordinated signal timing parameters as well as independent intersection control to adapt to changing traffic conditions. Recent interest in implementing adaptive control strategies as a means of accommodating highly variable and congested traffic conditions has led to several domestic installations.

In our country, the situation is totally different:

- There are no important UTMC systems implemented (exceptions are the UTOPIA for 41 tram line in Bucharest, and the future Bucharest Multisector UTC/PTM/CCTV project for around 100 junctions and 300 buses);
- The signalling infrastructure is heterogeneous, obsolete and frequent failures are reported;
- There is no a centralised concept in monitoring, control and repair operations for the road signalling infrastructure;
- There is no an updated database concerning the field facilities kept at an administrative authority (such as Street Administration) – so as at this moment, if a traffic signal or sign is damaged or missing, there will be a big delay (days) until the authority will be noticed and maintenance will be performed;
- The Street Administration required several times a centralised, integrated method/platform for field data acquisition/monitoring of the signalling infrastructure.

According to these reasons, it is undeniable that Bucharest's street signalling infrastructure needs an important review and improvement.

[5] Split Cycle Offset Optimisation Technique
[6] Urban Traffic Management Centers
[7] Sydney Coordinated Adaptive Traffic System
[8] Urban Traffic Optimisation by Integrated Automation

2 Designing an ITS platform for the road signalling infrastructure

For the signalling infrastructure it is very important to have an integrated platform for data acquisition and functional status monitoring. The main functions of such a platform should be:

a. Possibility for data collection concerning actual position of signalling elements (for this purpose, a method for easily collect geographic data information is to be developed – the best method for building a database with the accurate geographic position for all signalling elements is to use human operators, with GPS enabled PDAs and an instrument for remotely collect these information, such a GPRS data transmission;

b. Building and maintaining a database with geographic and functional information about the static and dynamic signalling elements in the road infrastructure;

c. Possibility for interconnection with future urban traffic management systems, such as the one that will be implemented in Bucharest;

d. Own data network (wireless or wired, optic fibber);

e. Possibility for automatic fault reporting from intelligent traffic controllers;

f. Possibility for automatic remote alarm issuing, in case of signalling infrastructure malfunction;

g. Users information system, with possibility to inform in real-time infrastructure operators, traffic participants etc. about signalling infrastructure malfunctions, traffic deviations etc.

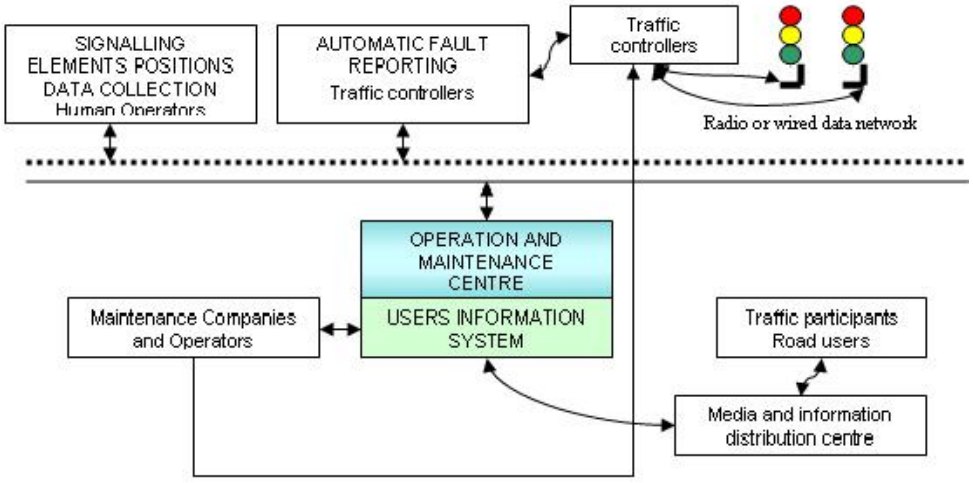


Figure 1: Functional architecture for the Optimised Signalling Infrastructure

As mentioned before, this architecture will better work if interconnected with an UTC system infrastructure. A possible global data transmission network is depicted in figure 2.

The UTC operators will handle the management of events of various kinds, for example:

- Forthcoming activities with traffic impact (works, demonstrations, sport meetings);
- Real-time events (accidents, incidents);
- Fault analysis and rectification – here the role of the Optimised Intrinsic Safety Signalling Infrastructure is determinant;

- Meteorological phenomena impact on the traffic, based on the weather reports.

All types of data should be converted to standard formats, and referenced to the road network description (infrastructure data base). Some particular challenges still need to be overcome, particularly with respect to detection. As Bucharest suffers from extreme temperature ranges, carriageway surface conditions can be poor.

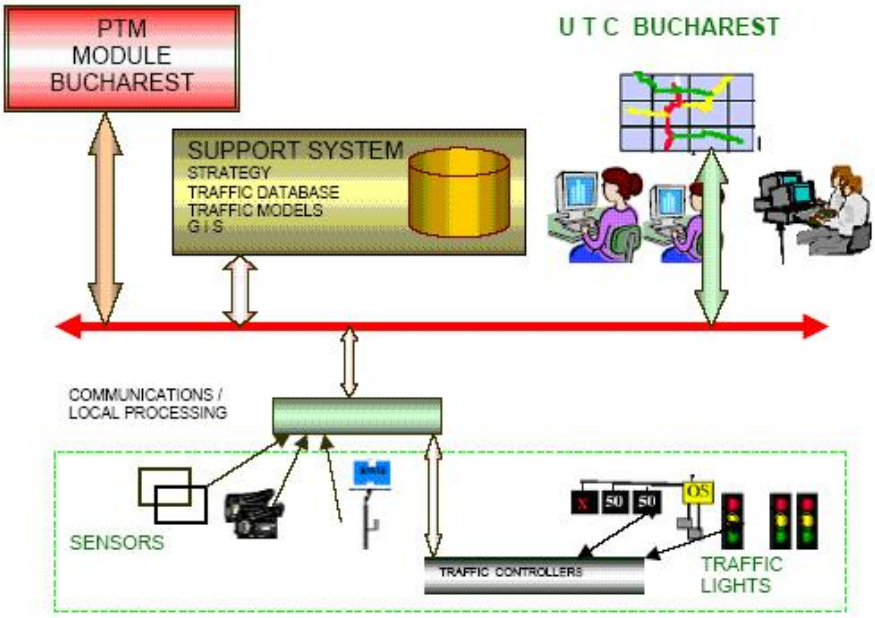


Figure 2: Possible system interconnection architecture for a combined signalling monitoring / UTC system

It must be emphasized that the integration of these elements of work should be in the forefront of the future traffic automations in Bucharest. This process should take into account the state-of-the-art technologies and the existing standards in the field of Intelligent Transport Systems. The architectures of the intelligent traffic signalling infrastructure, the UTC and PTM systems should be designed using the newest methodology, for example the Object Oriented approach. As consequence, the UTC and PTM systems should be "building blocks", should have open and scalable architectures and should be fully interoperable with other new ITS systems or modules. Moreover, these systems should implement all the XML-based data exchange standards for road traffic and public transport information, like DATEX, TRIDENT and OTAP. This is highly important as in the near future all the European integrated infrastructures with UTC and PTM systems will consist also an integrated part in a distributed national ITS architecture. These systems will act as the main components of the future ITS Euro-region planned architecture that is to be developed in Eastern Europe in the next coming years.

3 Designing a dedicated signalling infrastructure database and maintaining it

The database for the integrated system should offer to the users and IT professionals' powerful and familiar instruments, reducing the complexity for designing, implementing, managing and using the data along with analytical applications, both on mobile and fixed platforms. Among the requirements for such a competitive database, there should be:

- *Relational database:* the database developed should have a safe, reliable, scalable and highly available engine, with support for both structured and unstructured data;
- *Replication services:* in order to correctly operate with distributed or mobile data processing applications, the built database should be able to replicate the data, to use secondary data stores for enterprise reporting solutions and the possibility to integrate with heterogeneous existing, external databases;

- *Alerting services:* the database should comprise advanced capabilities for developing and implementing of the scalable applications, that are able to offer updated personalised information, for a large variety of mobile and fixed interconnected devices;
- *Integration services:* Capabilities for information extraction, transforming and loading for data warehousing and locally integration;
- *Analyse services:* The database should be able to perform Online Analytical Processing (OLAP), for rapid and elaborated analysis of large data sets, that use multidimensional storage;
- *Reporting services:* the database should be able to offer a complete solution for designing, management, and reporting, in several formats: on paper and interactive, web-based;
- *Management instruments:* the database server should include integrated management systems for administration and advanced data optimisation, along with close integration with other specific instruments;
- *Developing instruments:* The database server should be empowered with integrated developing instruments for the database engine, for extracting, transforming and loading of data etc.

a. **Specific requirements for the integrated optimised signalling infrastructure**

Traffic management systems by definition contain functions for the detection and selection of vehicles and functions associated with the granting of priority. A very important characteristic of the communications network for these systems is that the delay for any operational data must be strictly controlled, in order that the system operates in real time. If the maximum communication delay time is over passed, then the status change at traffic signals, operated by the system, might not have the desired effects (i.e. granting priority to public transport or emergency vehicles in real time). Furthermore, several traffic disturbances that might occur could increase the degree of congestion, making traffic signal queues and transit times longer. The communications network is a crucial element in an optimised signalling infrastructure, as the whole sensor network relies also on real time transmission of information.

3.1 Specific architectures for priority granting and fault management functions

The starting point in the process for granting selected vehicles priority is the location of a vehicle at a point and time within the controlled network. In systems which prioritise selected vehicles according to pre-determined criteria, information relating to the particular properties of an individual vehicle may be collated with other data such as timetable information to form a decision on whether priority should be requested. This decision may take into account a forecast of the future location of the vehicle. The output from these functions is either a simple request for priority or a priority level for the vehicle concerned.

When the priority strategy is separate and interrupts the background strategy, then there is scope for a variety of architectures. It is possible, but not necessarily desirable for some or all of the priority functions to be logically separated from the background control function. Four groups of architectures are possible:

- Centralised UTC and priority functions, centralised fault reporting;
- Decentralised UTC and priority functions, centralised fault reporting;
- Centralised UTC with decentralised priority functions, centralised fault reporting;
- Decentralised UTC with centralised priority functions, centralised fault reporting.

As seen in the above classification, the fault reporting / management function should be always centralised, as this is very important for the good operational status of the system. In all cases, even if there is or there is not a wired communication network link to a management centre, there is a must to have a radio link (or GSM network link) for the fault monitoring function. Below are presented several diagrams showing these architectures. In figure 3 a complete wired, centralised architecture is depicted. The traffic controllers are interconnected and mastered by the Urban Traffic Control Centre; all fault reporting are transmitted via the traffic control network. In case there is not implemented a centralised solution, or there is only a part of the traffic controllers networked, while the rest are not yet introduced in the centralised system, the best architecture is a wireless one, as depicted in figure 4.

Usually, the reliability requirements are taken into account mainly in centralised traffic systems. With the second case, the reliability of the signalling infrastructure can be significantly improved, over imposing a system to monitor in a centralised manner all faults. Synchronisation for all system's signalling can be achieved via a GPS[9] clock signal, used also for *green waves* in traffic signalling.

3.2 Specific requirements for delays in communication

When using a centralised system with priority granting at traffic signals for special vehicles, the communication delay must be controlled and under a specific threshold. Otherwise, the vehicle can arrive in the junction and the phase of the signal change too late, increasing traffic congestion.

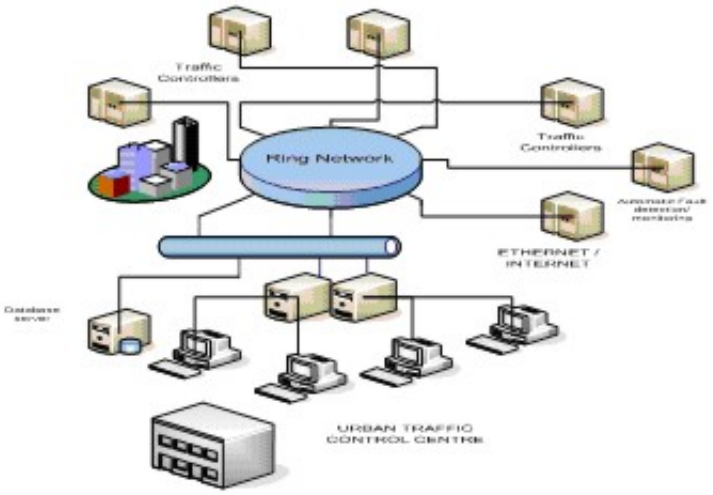


Figure 3: Centralised control, all functions

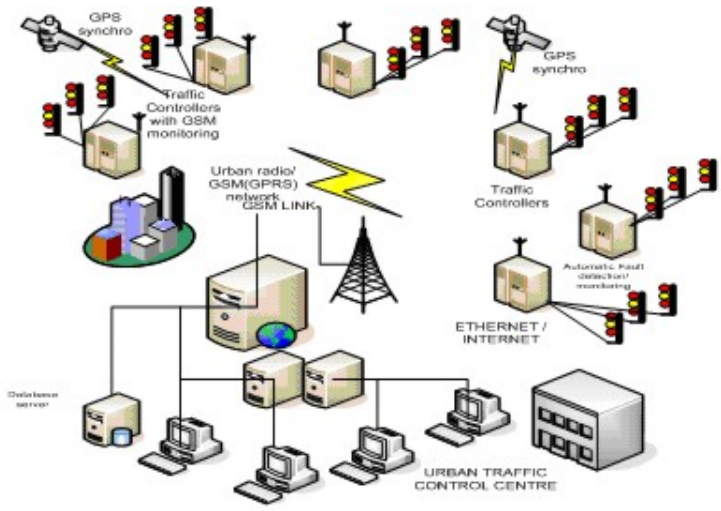


Figure 4: Decentralised UTC functions, centralised fault monitoring functions

[9]Global Positioning System

4 Conclusions

There is an important need for improving the intrinsic reliability for the urban traffic signalling infrastructure, as in many cities from our country the traffic signalling systems are not yet centralised, and not sufficiently monitored. Using an integrated platform for data acquisition/fault management can increase significantly the operational status of the traffic signals, contributing to the reduction of traffic incidents, congestion and environmental pollution.

References

[1] Minea, M., Grafu, F.D. – Telematică în transporturi. Noţiuni de bază şi aplicaţii. Ed. Printech, Bucureşti 2005, ISBN 973-718-336-3.

[2] UPB-CEPETET – Creşterea Siguranţei Intrinseci a Infrastructurii de Semnalizare Rutieră – Contract de cercetare ROSARO (ROmanian SAfe ROutes) CEEX 2005, RELANSIN.

[3] M. Minea, E. Catană, A. Withill, Gh. Udrişte, A. Grigorescu, R. Timnea – Integrated Adaptive Urban Traffic Control System with Public Transport Management System in the e-BISUT project. 12-th World Congress on ITS, San Francisco, California, USA – Nov. 2005, paper ID 2030.

[4] M. Minea, Gh. Stan, F.C. Nemţanu. *Incidence of New Telematic Systems for Treansports in Romanian Information Society.* Proceedings of ICCC 2004 (International Conference on Computers and Communications) – University of Oradea, Baile Felix, 27-29 Mai 2004, pp. 248-254.

Marius Minea, Florin Codruţ Neamţu
Politehnica University of Bucharest
Transports Faculty, Telematics and Electronics for Transports
Address: 313, Splaiul Independenţei, Building J, JF105
Sector 6 Bucharest 060042
E-mail: {mariusminea,fneamtu}@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 320-325

Developing an Usability Evaluation Module Using AOP

Grigoreta Sofia Moldovan, Adriana Mihaela Tarţa

Abstract: Usability is one of the most important features of an interactive system, because it is a measure of user performance and satisfaction related to a software system. There are many methods for usability evaluation but most of them are expensive and difficult to integrate and use during software system development. In this paper we present an usability evaluation module developed using AOP that is easy to integrate and use during usability evaluation.

Keywords: usability evaluation, AOP

1 Introduction

1.1 Usability

Usability is one of software system's qualities as defined in ISO 9126-1 [6]. In ISO 9241-11 [7] usability is defined as: *the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*, where *effectiveness* is the accuracy and completeness with which users achieve specified tasks, *efficiency* is the resources expended in relation to the accuracy and completeness with which users achieve goals, and *satisfaction* concerns the comfort and acceptability of use by end users. Many other definitions have been formulated along time for *usability*, but only few of them have been operationalized in order to be easy to use for software designers.

In [14], Shackel defines the usability of a system as *the capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and user support, to fulfill the specified range of tasks within environmental scenarios*. The definition is then operationalized by using four criteria: effectiveness, learnability, flexibility and attitude.

Schneiderman defines usability by means of five aspects: speed of performance, time to learn, retention over time, rate of errors and subjective satisfaction [15].

Nielsen has a similar approach to Schneiderman regarding usability which is defined by: efficiency, learnability, errors/safety and satisfaction [11].

In [5], Dix gives a more clear explanation of usability concept, using three categories of factors that determine usability: learnability, flexibility and robustness, and each category is described using usability indicators. The usability indicators are presented in Table 1.

Learnability	**Flexibility**	**Robustness**
Predictability	Dialog initiative	Observability
Synthetizability	Multi-threading	Recoverability
Familiarity	Task Migrability	Responsiveness
Generalizability	Substitutivity	Task conformance
Consistency	Customizability	

Table 1: Usability indicators

1.2 Usability Evaluation

Usability evaluation is an important part of user interface design that tries to identify and predict usability problems. Nowadays, there are several methods that are trying to evaluate the usability of a system. Some methods can evaluate only WIMP(windows, icons, pointer, and mouse) user interfaces (UI), other methods can evaluate only Web interfaces, and a few can be used to evaluate both WIMP and Web interfaces.

Different approaches have been used for evaluating usability: testing, inspection, inquirying, analytical modeling and simulation [8].

Independently of the approach used, most evaluation methods include three activities:

Capture collecting usability data, such as task completion time, errors, guideline violations, and subjective ratings;

Analysis interpreting usability data to identify usability problems in the interface;

Critique suggesting solutions or improvements to mitigate problems.

Many parts of these activities can be done automatically. The automation of these activities has the following advantages: reduced cost of usability evaluation, reduced need for evaluation expertise among individual evaluators, increased coverage of evaluated features, enables comparisons between alternative designs, and easy incorporation of evaluation within the design phase of user interface development, as opposed to after implementation [8].

1.3 Aspect Oriented Programming

Aspect oriented programming (AOP) is a new programming paradigm that addresses the issues of *crosscutting concerns* [9]. A crosscutting concern is a feature of a software system whose implementation is spread all over the system. Well-known examples of crosscutting concerns are logging and security. In order to implement a crosscutting concern, AOP introduces four new notions: *joinpoint*, *pointcut*, *advice* and *aspect*:

- A *join point* is a well-defined point in the execution of a program.
- A *pointcut* groups a set of join points, and exposes some of the values in the execution context of those join points.
- An *advice* is a piece of code that is executed at each join point in a pointcut.
- An *aspect* is a crosscutting type that encapsulates pointcuts, advice, and static crosscutting features. An aspect is the modularization unit of AOP.

The aspects are integrated into the system using a special tool called *weaver*.

Nowadays, there are extensions that support AOP for well-known programming languages (i.e. AspectJ for Java [4]) which are used in industry, too.

Section 2 presents a short description of the usability methods we have based our research on (section 2.1), an analysis of what should our module do (section 2.2), the design of our module (section 2.3) and some code examples (section 2.4). Conclusions and further work are presented in section 3.

2 Usability Testing Methods

We have chosen to develop a module to support usability testing methods.

2.1 Description

Usability testing with real participants is a fundamental usability evaluation method [11], [15]. It provides an evaluator with direct information about how people use computers and which are the problems of the interface being tested. During usability testing, participants use the system or a prototype to complete a predetermined set of tasks while the evaluator records the results of the participantsŠ work. The evaluator then uses these results to determine how well the interface supports usersŠ task completion as well as other measures, such as number of errors and task completion time [8].

2.2 Analysis

The activities that can be fully automated are capture and analysis. The critique activity can be automated (for example using an expert system), but it still requires human expertise. Our research focuses only on capture and analysis activities.

The capture activity consists of logging events and information like: pressing a button, selecting an option, completing, abandoning or failing a task, error occurences, navigating the menu, etc.

Logging is a well known crosscutting concern that should be designed and implemented using AOP as it spans across multiple modules. The logged information depends on the usability aspects we want to evaluate (i.e. user-performance, user satisfaction, task completion, etc.).

The analysis activity consists of analyzing the log files to compute usability metrics. The usability metrics that are usually computed are presented in Table 2.

Metric	**Usage indicator**
Time to complete a specific task	Performance time
Number of commands used	Performance time
Percent of task completed per unit time	Performance time
Relative time spent in physical actions	Performance time
Relative time spent in mental actions	Performance time
Number of tasks that can be completed within a given limit time	Performance time
Number of regressive behaviors	Memorability
Number of system features users can remember afterwards	Memorability
Time spent in errors	Errors
Percent or number of errors	Errors
Number of repetitions of failed commands	Errors
Percent of task completed	Task completion
Number of available commands not invoked	Task completion
Ratio of success to failures	Task completion
Number of runs of successes and failures	Task completion

Table 2: Usability metrics

The existing tools ([12], [10], [13]), that help evaluators, perform the analysis only after the user-UI interaction is ended, and some of them even require the UI to be developed in a special environment [12].

AOP gives an alternative to these tools, as the metrics can be computed during user-UI testing session, without the need to use log files. Additionally, it does not require the UI to be developed in a special environment. The only constraint is that we have to use the AO extension corresponding to the programming language used for UI implementation. There are extensions for a variety of programming languages: AspectJ for Java [4], AspectC for C [1], AspectC++ for C++ [2], AspectC# for C# [3] etc.)

2.3 Design

For our module we have chosen the following functionalities related to usability evaluation: to log errors (a log record contains the error message and the time of occurence), to capture a screen shot each time an error occurs, to compute the number of successfully completed tasks, the number of abandoned tasks and the number of failed tasks, and to compute the frequency of each error type. A task is completed when it is started, executed and there are no errors during the execution; a task is failed when it is started, executed, and there are errors during the execution, and a task is abandoned when it is started but it is not executed.

In order to design the usability evaluation module using AOP, we first have to decide which are the points from the program flow we are interested in. First we have to determine the point from where we start measuring, denoted by *entryPoint()*, and the point to which we stop measuring, denoted by *exitPoint()*. These will be two pointcuts in an aspect called *AppAspect*.

Afterwards, depending on the information we want to gather, we define new aspects. For example, if we want to log each error that a user performed when using the system, we have to define a new aspect *ErrorAspect*, that captures the points in the program flow that indicate errors, with a pointcut *error()*.

If we want to compute metrics related to tasks we define a new aspect *TaskAspect*, with two pointcuts: *startTask()* and *execTask()*. The *startTask()* pointcut captures the points from the control flow when a new task is started and the *execTask()* pointcut captures the points from the control flow when a task is executed.

An UML-like diagram for these aspects, with their pointcuts is shown in Figure 1.

The advices for each pointcut will contain the necessary computation that have to be done.

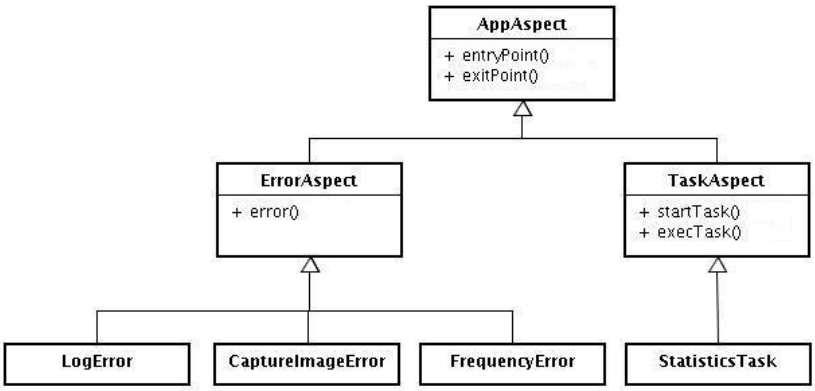


Figure 1: Module's aspects hierarchy

2.4 Implementation

To test our module, we have chosen a small family budget application implemented in Java. We have used AspectJ [4] as AOP extension. The application has the following functionalities: adding an expense/income, viewing expenses/incomes for a period of time and computing the balance for a period of time. An income/expense has a date, a value, a category, and a family member. The functionalities are available using menus or toolbar buttons.

The application uses the static *showError(String title, String msg)* method of the *MessageWindow* class to display error messages. The *error()* pointcut that captures errors is shown in Table 3.

```
abstract aspect ErrorAspect extends AppAspect{
  pointcut error(String msg):
    execution(* MessageWindow.showError(String,String)) && args(.., msg);
}
```

Table 3: *ErrorAspect* code.

The source code of the *LogError* aspect is shown in Table 4. The source code that captures screenshots and computes errors frequency is almost the same.

3 Conclusions and further work

We have designed a module for an automatic usability evaluation method based on user-testing. The advantages of this approach are:

- the source code for usability data gathering is kept in one place: the usability evaluation module;
- the software system modules do not need to be modified in order to obtain usability data;
- the usability evaluation module is easy to integrate with the other modules;
- the usability evaluation module is easy to plug-in and out of the system. In order to plug it in we just have to call the *weaver* tool with the source code or the binaries of the system;
- the usability module can be easily adapted to other software systems, we just have to modify the pointcuts of the aspects;
- if we want to add or remove some data, only the usability evaluation module has to be modified, even though the needed data is spread all over the software system.

```
aspect LogError extends ErrorAspect {
    private LogWriter lw;
    before() : entryPoint() {
        //initializes the log writer
    }
    before(String msg) : error(msg) {
        lw.log(msg,LogWriter.ERROR);
    }
    before() : exitPoint() {
        //closes the log writer
    }
}
```

Table 4: *LogError* code.

One disadvantage of this approach is that the developer of the usability evaluation module needs to have an indepth knowledge of some parts of the software systems, in order to be able to define the pointcuts. This disadvantage can be overcome if the developer of the evaluation module is cooperating with the software system developers.

As further work, we intend to develop a tool that can be used to write the variant parts of the module (the pointcuts) to adapt them for various applications written in Java and then to automatically generate and integrate the module within the software system.

References

[1] AspectC: AOP for C. http://www.cs.ubc.ca/labs/spl/projects/aspectc.html.

[2] AspectC++ home. http://www.aspectc.org/.

[3] AspectC#: Intro. http://www.dsg.cs.tcd.ie/index.php?category_id=168.

[4] AspectJ Project. http://eclipse.org/aspectj/.

[5] Alan Dix, Janet Finley, Gregory Abowd, and Russell Beale. *Human-computer interaction (2nd ed.)*. Prentice-Hall, Inc., 1998.

[6] ISO 9126, Software product evaluation - Quality characteristics and guidelines for their use.

[7] ISO 9214-11, Ergonomic requirements for office Work with VDT's - Guidance on usability.

[8] Melody Y. Ivory and Marti A Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computings Surveys*, 33(4):470–516, 2001.

[9] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-Oriented Programming. In *Proceedings European Conference on Object-Oriented Programming*, volume 1241, pages 220–242. Springer-Verlag, 1997.

[10] M. Macleod and R. Rengger. The development of drum: A software tool for videoassisted usability evaluation. In *Proceedings of the HCI Conference on People and Computers VIII*, pages 293–309. Cambridge University Press, 1993.

[11] Jakob Nielsen. *Usability Engineering*. Academic Press, 1993.

[12] Dan R. Olsen and Bradley W. Halversen. Interface usage measurements in a user interface management system. In *UIST '88: Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, pages 102–108, New York, NY, USA, 1988. ACM Press.

[13] M. Rauterberg. *From novice to expert decision behaviour: A qualitative modeling approach with Petri nets*. Elsevier Science Publishers, 1995.

[14] Brian Shackel and S. J. Richardson, editors. *Human factors for informatics usability*, chapter Usability - Context, Framework, Definition, Design and Evaluation. Cambridge University Press, 1991.

[15] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., 1986.

Grigoreta Sofia Moldovan, Adriana-Mihaela Tarţa
Babeş-Bolyai University
Department of Mathematics and Computer Science
Address: Str. Mihail Kogalniceanu nr. 1, RO-400084 Cluj-Napoca
E-mail: {grigo,adriana}@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 326-331

The Experimental Analysis of the Impact of the "Nogood Processor" Technique on the Efficiency of the Asynchronous Techniques

Ionel Muscalagiu, Vladimir Creţu, Manuela Pănoiu, Caius Pănoiu

Abstract: One of the characteristics of the algorithms of asynchronous search is given by the occurrence of nogood values during solution search. Nogood values show the cause of the failure and their incorporation as a new constraint will teach the agents not to repeat the same mistake. In this article we will continue to adapt the nogood processor technique to the AWSC technique, and we will analyze experimentally the benefits to the efficiency. This technique consists in storing the nogood values and further use the information given by nogoods in the process of selecting a new value for the variables associated to agents.
In the article we analyzed the distribution of nogood values to agents and the way to use the information stored in the nogood, what we will call the nogood processor technique. We analyzed the situations in which a centralized nogood processor is used and the situation in which the nogoods are distributed to more nogood processors manipulated by certain agents. We analyzed the benefits that the nogood processor technique brings to enhancing the performance of the AWCS technique, by identifying a few ways of distributing their efficiency.

Keywords: constraints, agents, nogood messages.

1 Introduction

The constraint programming is a model of the software technologies, used to describe and solve large classes of problems as, for instance, searching problems, combinatorial problems, planning problems, etc. Lately, the A.I community has show great interest towards the distributed problems that are solvable through modeling by constraints and agents. The idea of sharing various parts of the problem between agents that act independently and that collaborate between them using messages, in the order to find a solution, has proven useful, as it has lead to obtaining a new modeling type called Distributed Constraint Satisfaction Problem (DCSP) [3].

In the distributed constraint satisfaction area, the asynchrounous weak-commitment search algorithm (AWCS) [3], plays a fundamental and pioneer role among algorithms for solving the distributed CSPs. The algorithm is remarked for the suffering of an explosion of the nogood values, but, by dynamically changing the agents' order, it is an efficient algorithm because of its number of cycles.

The non - restriction for recording the nogood values could become, in certain cases, impracticable. The main reason is that the storing of nogood values excessively consumes memory and could lead to lowering the memory that has been left. Typically, the number of nogood values grows along with the number of conflicts - in the worst case this growth could be exponential in the number of variables. Another unpleasant effect of storing a large number of nogood values is tied to the fact that the verification of the current associations in the list of nogood values that are stored becomes very expensive, the searching effort removing the benefits brought by storing the nogood values. These elements are analyzed as targeting to see if this nogood processor technique brings benefits for efficiency.

In [1] Armstrong and Durfee builds a new asynchronous technique in which the problem's solution is divided into epochs. There is a central agent responsible for the start of the searching process and a nogood processor that keeps information on the nogoods occurred. When a nogood is discovered , the variable assignments causing it and the IDs of the agents involved are sent to the nogood processor, which saves the nogood. Later, when an agent has found a tentative assignment for its variables, it consults the nogood processor to make sure that its assignment along with any known assignments of higher priority agents do not constitute a nogood.

In [2] is presented a first way of applying the technique with a nogood processor (induced in [1]) to the AWCS technique, but without investigating the efficiency of the derived techniques obtained. There the technique nogood processor is combined with the technique nogood learning, without analyzing the impact of applying only the technique nogood processor. In this article we will continue to adapt the nogood processor technique to the AWSC technique, and we will analyze experimentally the benefits to the efficiency. We will experimentally show that there is a way of distributing the nogood processors, that will bring significant for the efficiency enhancements, which will reduce the costs of communications without being affected by the new costs that occur for the verification of nogoods.

2 The Framework

In order to do this analysis of the nogood processor impact, in this paragraph we will present some notions known from the IT literature relative to the DCSP modeling and the asynchrounous weak-commitment search algorithm [3], .

Definition 1 (CSP model). The model based on constraints CSP-Constraint Satisfaction Problem, existing for centralized architectures, consists in:

-n variables $X_1, X_2.X_n$, whose values are taken from finite, discrete domains $D_1, D_2, , D_n$, respectively.

-a set of constraints on their values.

The solution of a CSP supposes to find an association of values to all the variables so that all the constraints should be fulfilled.

Definition 2 (The DCSP model). A problem of satisfying the distributed constraints $(DCSP)$ is a *CSP*, in which the variables and constraints are distributed among autonomous agents that communicate by transmitting, messages.

In this article we will consider that each agent A_i has allocated a single variable x_i.

Definition 3 (the list agent-view). The list agent-view of an agent A_i is a set with the newest assignments received by the A_i agent for distinct variables.

Definition 4 (the nogood list). The Nogood list is a set of assignments for distinctive variables for which looseness was found.

The AWCS algorithm [3], is a hybrid algorithm obtained by the combination of asynchronous backtracking algorithm (ABT) with WCS algorithm, which exists for CSP. It can be considered as being an improved ABT variant, but not necessarily by reducing the nogood values, but by changing the priority order. The AWCS algorithm uses, like ABT, the two types of ok and nogood messages, with the same significance.

For a better understanding of the way the nogood processor technique applies, we will present more information related to the behavior of an agent A_i, for the AWCS [3], algorithm. When an agent A_i receives an ok? message, it updates its agent view list and tests if a few nogood values are violated. A very important thing is connected to agent testing only the nogood values that have a greater priority than x_i (the authors use the maximal priority nogood term for these values). As a matter of fact the priority of a nogood value is defined as the lowest priority of the nogood variables, excepting x_i. As a conclusion, a generic agent A_i can have the following behavior:

- If no higher priority nogood value is violated, it doesn't do anything.
- If there are a few higher priority nogood values that have inconsistent values and these values could be eliminated by changing the x_i value, the agent will change this value and will send the ok? Message. If it has to choose between more values, it will select that value that minimizes the inconsistencies in the inferior priority nogood values (a nogood value of inferior priority is the one in which its priority is inferior to the x_i priority).
- If a few higher priority values are inconsistent and this inconsistence can not be eliminated, the agent creates a new nogood messages outside the agent view list and sends a nogood message to each agent that has variables in nogood. Than the agent increases the priority of x_i, by changing the x_i value with another value that minimizes the inconsistencies number with all the nogood values and sent the ok? message. If the new nogood is identical to the previous nogood value, than the agent will do nothing.

We must underline a certain behavior of A_i agent, specific to the AWCS algorithm: when a nogood message is received, the agent adds the nogood value to its nogood group and executes a verification of the inconsistencies for nogood. If the new nogood has also an unknown variable, the agent needs to receive from the corresponding agent the value of the variable that's been cared for. Unfortunately, the information from nogood is not completely used. It is about the case of attributing a new value for the variable associated to the agent. It is possible that the nogood values contain a reference to this value, that implying the attribution to have appeared more as inconsistent. The use of this information will be the basis of the nogood processor technique construction.

3 Adapting the "NOGOOD" Processor Technique

3.1 Implementing the technique "nogood processor"

In [2] is presented a first way of applying the technique with a nogood processor to the AWCS technique, but without investigating the efficiency of the derived techniques obtained. The adaptation and the application of the nogood processor technique for the AWCS technique lead to the identification of some answers to the problems occurred: how do we store the nogood values and where, how is the storing and evaluation of nogood values distributed? Another very important problem was to see if all agents (or a part of them) appeal the nogood processor, so as there will be no chance that the costs necessary for evaluating the stored nogoods would outrun the benefits brought by the nogood processor technique.

We consider each agent to have access to the results of its own nogood processor or of a centralized nogood processor (typically using messages). More, each agent sends (stores) nogood values that it has received to the associated nogood processor. Therefore, when using a single processor, the treating procedure for nogood messages will have to store in a shared memory zone or to send the nogood values to the nogood processor. We will assume, further, that these nogood values are stored in the *nogoods-store* list. The information stored by each nogood processor will be used in searching a new value for each variable cared for by the agent (in AWCS algorithm). For this, each nogood processor will verify (asked by an agent) through its subroutine
check-inconsistent-value-nogood-processor, if the value selected by the agent had no previous existence associated with the higher priority agents values. In figure 1 we have the checking routine for the inconsistence of a new value. This routine
(check-inconsistent-value-nogood-processor) is applied to the AWCS algorithm.

```
function check-inconsistent-value-nogood-processor [A_i]
    foreach Nogood ∈ nogoods-store do
        foreach x ∈ Nogood with the current priority from current-view
                * bigger than the agent's A_i
          pos ⟵ position x in Nogood
          if x != item pos current-value
              return consistent
           endif
        end do
        if current-value != item A_i in nogood
           return consistent
        endif
        end do
       return inconsistent
end procedure
```

Figure 1: The Procedure check-inconsistent-value-nogood-processor

To increase the efficiency of the nogood processor, when receiving an ok type message, informing about setting up a new value from an agent, the nogood processor verifies the value for the higher priority agents. Thus, if that combination has ever existed as nogood, we try searching a new value for the current agent.

The answer to a question was seeked which consisted in identifying the way of distributing the nogood messages to processors. In this article are proposed two solutions of distributing the nogoods to the processors:

- a single nogood processor operated by an central agent. The proposed variants in this study based on the usage of a single "nogood processor" will be noted $AWCS_{2k}$.
- many "nogood processors" distributed to certain agents (based on some rules that will be presented). The variants with many nogood processors will be noted $AWCS_{3k}$.

3.2 Versions of AWCS with a single "nogood processor"

The first solution consists in adapting the AWCS tehnique such as to store centralized the nogood values. These versions will be noted $AWCS_{2k}$. The basic technique from [3], will be noted $AWCS_1$.

Each agent in the moment of receiving of a nogood, transmit sit to a central agent that has a centralized nogood agent processor. That agent will stock the value received in a list nogood-store (all the "nogood processors" save only those new values, eliminating the copies). The information from here will be used later in the process of searching of a new value. Thus, the procedure
check-agent-view will select a new value consistent with the list agent-view and with the list of nogood values stocked by the "nogood processor". Different from the basic variant of AWCS, the new variant AWCS will select a new value if, in addition, the procedure
check-inconsistent-value-nogood-processor will return the consistent value. Also, the procedure for checking the nogood values will be called, also in the Backtrack procedure, in the moment of selecting a new value.

For that first solution with a single nogood processor, several versions are proposed. The versions differ in the way that the agents that appeal the "nogood processor" are identified, and in the way that the identification of the agents for which is made the comparison of the combination of values. A first version implies the applying of the procedure
check-inconsistent-value-nogood-processor presented in figure 1, version noted $AWCS_{21}$. The checking procedure of the nogood values checks this thing only for the agents that have the current priority from current-view greater than of the agent A_i. This version will be the basic variant with a centralized "nogood processor".

A second version proposed, noted $AWCS_{22}$, was obtained by adapting the procedure
check-inconsistent-value-nogood-processor such as it checks the values of higher priority, priority holded in the moment of stocking the nogood values. In other words, the identification of the agents of higher priority than the agent Ai, isn’t done using their current value (that is in current-view), but is done using the value of the priority stocked by the nogood processor. The corresponding modification was marked with * in the checking procedure. The values of these agents are checked if they existed before in that combination.

The next versions are based on identifying the agents that will call the associated "nogood processors". Many versions are proposed that differ by that that not all the agents use the information from the nogood processors. The third version proposed, noted $AWCS_{23}$, is obtained starting from the variant $AWCS_{21}$ such as the call of procedure check-inconsistent-value-nogood-processor be done by all agents, excepting that of the highest priority from the neighbors of that moment. A forth version, noted $AWCS_{24}$, is obtained from the variant $AWCS_{21}$ such as the call of the procedure check-inconsistent-value-nogood-processor will be done only by the agent with the highest priority from the neighbors (current priority from current-view respectively the old one stocked).

A last version, noted $AWCS_{25}$, is obtained from the variant $AWCS_{22}$ such as the checking be done only by the agent with the highest priority from the neighbors (current priority from current-view respectively the old one stocked). That last version based on a centralized nogood processor is remarked by the fact that the identification of the agents of higher priority is done relatively to their priority from the moment of stocking and by the fact that not all the agents call the nogood processor.

3.3 Versions of AWCS with many "nogood processors"

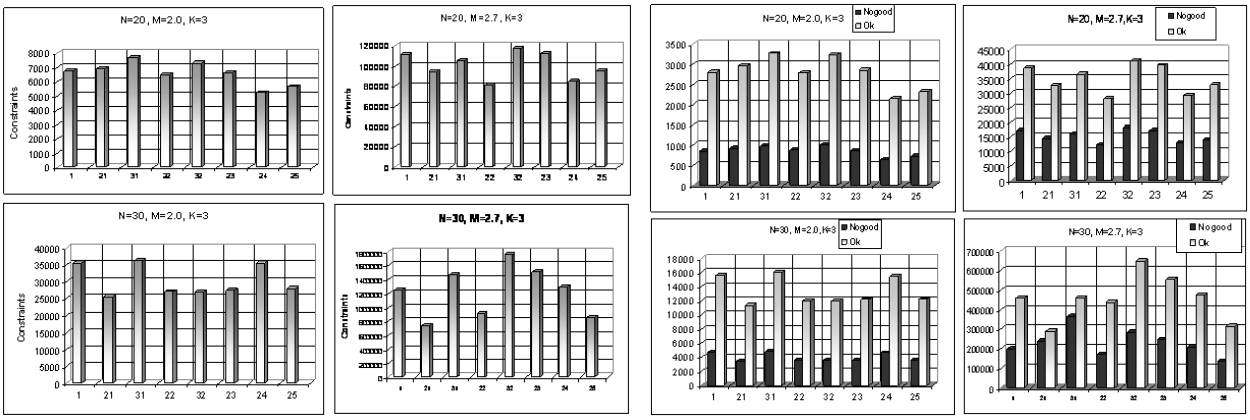
A second solution is based on distributing the nogood values to many "nogood processors", one for each agent. Those versions will be noted $AWCS_{3k}$. In the moment of receiving a nogood value, it is stocked only by the associated nogood processor. The nogood value isn’t any more transmitted to a central agent, but is stocked locally. As in the case of the $AWCS_{2k}$ versions, the information from here will be later used in the process of search.

Selecting a new value supposes checking also the stocked nogood values. The check is done similar to the solution with a single nogood processor, obtaining two versions noted $AWCS_{31}$ and $AWCS_{32}$ (the last one uses the way of identifying the agents of higher priority using the older priorities). The version $AWCS_{31}$ will represent the basic variant with a distributed "nogood processor".

4 Experimental results.

In this paragraph we will present our experimental results, obtained by implementing and evaluating the asynchronous techniques we introduced. In order to make such estimation, we implemented these techniques in NetLogo 2.0, a distributed environment, using a special language [4], [5].

The asynchronous techniques were applied to a classical problem: the problem of colouring a graph in the distributed versions. For the problem of graph colouring we took into consideration two types of problems -(we kept in mind the parameters n - number of knots/agents, k-3 colours and m - the number of connections between the agents). We evaluated two types of graphs: graphs with few connections (called sparse problems, having m=n



(a) Constraints checks

(b) nogood and ok messages

Figure 2: Comparative study for the AWCS versions-(Distributed n-Graph-Coloring Problem)

x 2 connections) and graphs with a special number of connections, known to be difficult problems (called difficult problems and having m=n x 2.7 connections). For each version we carried out a number of 100 trials, retaining the average of the measured values (for each class 10 graphs are generated randomly, for each graph being generated 10 initial values).

We counted the number of messages (which means the quantity of ok and nogood messages) and the number of constraint checks for obtaining each solution. The evaluations have been made for each technique presented. The evaluations had certain particularities due to the NetLogo medium. The NetLogo is a programming medium with agents that allows implementing the asynchronous techniques [4], [5], but has certain particularities related to asynchronous work with agents. The agents work with the specific command "ask". A command like this will allow launching the work routines with the messages. Each agent works asynchronously with the messages, but at the end of a command's execution there is a synchronization of agents' execution, synchronization that particularizes, in a way, the implementations being used.

In the figure 3(a) and (b) we presented graphically the experimental results for the 8 version of AWCS, with regard to the number of constraints and messages stream.

The first measuring unit for the analyzed asynchronous techniques performances was the number of constraint checks. As known, the verified constraints quantity evaluates the local effort given by each agent. By analyzing the graphics from the figure 3(a), where we represented graphically the number of constraint checks for obtaining the solution, we notice the fact that the nogood processor technique was effective on the basic AWCS technique performances. Among the analyzed versions we notice the performances given by the $AWCS_{22}$, $AWCS_{24}$ and $AWCS_{25}$ versions. The three versions had a smaller number of constraints for both problem classes: sparse and difficult problems. Still there is to be remarked the most difficult problem (n=30,m=2.7), the best behavior the $AWCS_{25}$ version had, when the manipulation of the nogood processor was made only by the agent with the highest priority among the neighbors, comparisons being made with respect to the old priorities.

In the last graph, figure 3(b), we presented the experimental results and the comparative study for the 8 versions, but with respect to the message stream. We counted the messages stream (which means the quantity of ok and nogood messages). It must be stressed that we counted only the nogood messages changed by agents, and we didn't calculate the messages necessary to send the nogood values to the nogood processors. The previous observations, were made during the evaluation with respect to the number of constraints stay in use, remarking the performances of the $AWCS_{22}$, $AWCS_{24}$ and $AWCS_{25}$ techniques. By analyzing the experimental results obtained, one can notice that the best results were obtained for the versions derived from the $AWCS_{21}$ case, a centralized nogood processor. In these cases there have been better results for performances, relative to the basic version AWCS.

5 Summary and Conclusions

In this article we tried the adaptation of the nogood processor technique for the AWCS. When a nogood is discovered during the solution searching process, this value is sent and stored by a nogood processor. Later, when

the search for a new value for a certain agent is tried, it is supplementary verified if that association (assignment) along with the values of the higher priority agents didn’t exist and, therefore, it is eliminated from the searching tree.

It is proposed a solution of distributing the nogood processors to agents, in regard to the order of the agents, with the purpose of reducing the searching and storing techniques. We experimentally analyzed the benefits the nogood processor technique brings to enhancing the performances of the AWCS technique, by identifying few distribution means that bring enhancements to the efficiency.

We analyzed more versions obtained by distributing the nogood values to more nogood processors for each agent. For identifying the higher priority agents we took into account the current priority of the agent (being in current-view) and the old priorities stored in the nogood-store lists.

The more performing versions have been obtained by distributing the nogood values just to the agent with the highest priority among the neighbors (reported to the current priority or the old one, stored). Enhancements of the performances have also been obtained in the case of the centralized nogood with old priority.

The evaluations have shown a reduction of the message stream, of the number of verified constraints for obtaining the solution. By distributing the nogood values and the nogood processors we obtained enhancements of performances.

References

[1] Armstrong, A. and E. Durfee (1997). *Dynamic Prioritization of Complex Agents in Distributed Constraint Satisfaction Problems*. In. Proceedings of the 15th IJCAI, Nagoya, Japan, 620-625.

[2] Muscalagiu I., V. Cretu. *Improving the Performances of Asynchronous Algorithms by Combining the Nogood Processors with the Nogood Learning Techniques*. Journal "INFORMATICA", Lithuania, 2006, Vol. 17, Nr 1.

[3] Yokoo, M., E.H. Durfee, T. Ishida, K. Kuwabara (1998). *The distributed constraint satisfaction problem : formalization and algorithms*. IEEE Transactions on Knowledge and Data Engineering 10 (5).

[4] Wilensky, U. (1999). *NetLogo*. http://ccl.northwestern.edu/netlogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

[5] *** *MAS Netlogo Models*. http://ccl.northwestern.edu/netlogo/models/community/, http://jmvidal.cse.sc.edu/netlogomas/.

Ionel Muscalagiu, Manuela Pănoiu, Caius Pănoiu
The Politehnica University of Timisoara
The Faculty of Engineering of Hunedoara
Address: 5, Revolutiei, Hunedoara, Romania
E-mail: {mionel,m.panoiu,c.panoiu}@fih.utt.ro
Vladimir Cretu
The Politehnica University of Timisoara
The Computers Science and Engineering Department Timisoara
Address: 2, V.Parvan St., Timisoara, Romania
E-mail: vcretu@cs.utt.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 332-335

Recognizing Dart-Free Graphs

Elena Nechita, Mihai Talmaciu, Gloria Cerasela Crişan

Abstract: The weakly decomposition of a graph that is the partition of the set of vertices in three classes A, B, C such that A induces a connected graph, and C is totally adjacent to B and totally nonadjacent to A. This paper presents a recognition algorithm of dart-free graphs, using the weakly decomposition.

Keywords: Dart graph, weakly decomposition, recognition algorithm.
AMS(2000):05C99.

1 Introduction

Let $G = (V,E)$ be a graph. In different problems from the theory of graphs, particularly in the building of some recognition algorithms, frequently appears a type of partition of the set of vertices in three classes A, B, C such that A induces a connected graph, and C is totally adjacent to B and totally nonadjacent to A. The introduction of the notion of weakly decomposition (C.Croitoru, E. Olaru and M. Talmaciu [3] , see also [5]) and the study of its properties allows the building of recognition algorithms of *dart*-free graphs.

2 Notations and fundamental definitions

All graphs in this work are undirected, with no loops or multiple edges. A graph is denoted $G = (V,E)$. $G(X)$ denotes the subgraph induced by vertex set X, and for $A \subset V$, $G-A = G(V-A)$. The *neighborhood* of a vertex x is $N_G(x) = \{y \neq x | xy \in E\}$, denoted $N(x)$ when there is no ambiguity. The neighborhood of a set of vertices A is $N(A) = (\cup_{x\in A} N(x)) - A$. The complement of the graph G is denoted by $\overline{G}$ and the non-neighborhood of a set of vertices A is $\overline{N}_G(A) = \{y \in V - A | xy \notin E, \forall x \in A\}$. A *clique* is a set of pairwise adjacent vertices, and is said to induce a *complete* subgraph. The maximum size of a clique in G is the *clique number* of G and is denoted $\omega(G)$. If k is a positive integer, a *k-coloring* of G is any assignment $c : V \to \{1,...,k\}$ with the property that for each $i \in \{1,...,k\}$ the set $c^{-1}(i) = \{v | v \in V, c(v) = i\}$ is a stable set in G, that is a set of mutually non-adjacent vertices. The least possible number k of colors (the set $S_i = c^{-1}(i)$ is called the color class i of the coloring c) for which a graph G has a k-coloring is called the *chromatic number* of G and is denoted $\chi(G)$. A K_n is a complete graph on n vertices, and a P_n is a chordless path on n vertices. The maximum size of a stable set in G is denoted $\alpha(G)$. If G is a graph with $\omega = \omega(G)$ and $\alpha = \alpha(G)$, a k-stable set or a k-clique (k is a positive integer) will mean a clique or a stable set of size k. *Connectivity* $k(G)$ of a graph G is the minimum number of vertices whose removal from G results is a disconnected graph or the trivial graph. A graph G is said to be *n-connected*, $n \geq 1$, if $k(G) \geq n$. An vertex v of a graph G is called a *cut-vertex* of G if $k(G-v) \geq k(G)$; thus, a vertex of a connected graph is a cut-vertex if its removal produces a disconnected graph. A *cutset* in a graph G is any set C of vertices such that $G-C$ is disconnected. A *star-cutset* is a cutset C having a vertex adjacent with all remaining vertices of C. Let $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ be two graphs. We note with G_1+ G_2, K_2-join of the graphs G_1 and G_2, that means, the graph obtained from K_2 by substituting its vertices with G_1 and G_2 and any vertex from G_1 is adjacent to any vertex from G_2. The graph G_1 +G_2 will be called the sum of graphs G_1 and G_2. A set A of vertices is totally adjacent with a set B of vertices ($A \cap B = \emptyset$) if ab is edge, for any a vertex of A and any b vertex of B (we note with $A \sim B$).

3 Basic properties

In this section we recall the notions and the results establish in [5] necessaries in the next section. For this we define the notion of weakly component, we establish its existence and we give a characterization for the weakly decomposition of a graph.

Definition 1. *Let G=(V,E) be a graph. A set of vertices, A, is called weakly set if $N_G(A) \neq V - A$ and the induced subgraph by A is connected. If A is a weakly set, maximal in relation to the inclusion, the subgraph induced by A is called weakly component. For simplification, the weakly component G(A), will be noted with A.*

The name of weakly component is justified by the next result.

Theorem 1. *Any connected and incomplete graph G=(V,E) admits a weakly component A such that* $G(V-A) = G(N(A)) + G(\overline{N}(A))$.

Theorem 2. *Let G=(V,E) be a connected and incomplete graph and* $A \subset V$. *Then A is the weakly component of G if and only if* $G(A)$ *is connected and* $N(A) \sim \overline{N}(A)$.

Definition 2. *A partition* $(A, N(A), V - A \cup N(A))$, *where A is a weakly set, is called weakly decomposition of graph G in relation to A. We call A, the weakly component,* $N(A)$ *the minimal cutset, and* $V - N(A)$ *the remote set.*

Theorem 3. *If* $G = (V,E)$ *is a connected and incomplete graph then the set of vertices V admits a weakly decomposition* (A,B,C) *such that* $G(A)$ *is a weakly component and* $G(V-A) = G(B) + G(C)$.

The Theorem 2 provides a polynomial algorithm for building a weakly decomposition for an incomplete and connected graph.

Algorithm for the weakly decomposition of a graph
Input: A connected graph with at least two nonadjacent vertices, $G = (V,E)$.
Output: A partition $V = (A,N,R)$ such that $G(A)$ is connected, $N = N(A)$, $A \not\sim R = \overline{N}(A)$.

begin
$A :=$ any set of vertices such that
$A \cup N(A) \neq V$
$N := N(A)$
$R := V - A \cup N(A)$
while ($\exists n \in N$, $\exists r \in R$ such that $nr \notin E$) *do*
$A := A \cup n$
$N := (N - \{n\}) \cup (N(n) \cap R)$
$R := R - (N(n) \cap R)$
end
One can observe that $[A]_G$ is connected, $N = N_G(A)$, $R \neq \emptyset$ is an invariant of the algorithm.

4 The algorithm

In this section we establish the algorithms of recognition for the class of $dart - free$ graphs.

A $dart$ is the graph with vertices *u*, *v*, *w*, *x*, *y* and edges *uv*, *vw*, *uy*, *vy*, *wy*, *xy*; a graph is called $dart - free$ if it has no induced subgraph isomorphic to a $dart$.

In [1] L. Babel, A. Brandstadt and V. B. Le present a polynomial-time algorithm which solves the reconstruction problem for BCD-free graphs (that are the graphs containing no induced copy of a banner, a chair, or a dart). A *banner* is the graph with vertices *u*, *v*, *w*, *x*, *y* and edges *uv*, *vw*, *uy*, *wy*, *xy*. A *chair* is the graph with vertices *u*, *v*, *w*, *x*, *y* and edges *uw*, *vw*, *wy*, *xy*.

In [2] V. Chvatal, J. Fonlupt, L. Sun, A. Zemirline present a polynomial-time algorithm to recognize dart-free Berge graphs. A graph *G* is called Berge graph if neither *G* nor its complement contains a chordless cycle whose length is odd and at least five.

In [4] C. Linhares Sales and F. Maffray present a polynomial-time algorithm to recognize dart-free perfectly contractile graphs. An *even pair* is a pair of vertices such that every chordless path between them has even length. A graph is perfectly contractile if every induced subgraph has a sequence of even-pair contractions that leads to a clique.

In what follows, we give a characterization of a $dart - free$ graph.

Theorem 4. *Let* $G = (V,E)$ *be connected with at least two nonadjacent vertices and* (A,N,R) *a weakly decomposition with A weakly component. G is* $dart - free$ *if and only if we have:*
(a) $G - A$ *and* $G - R$ *are* $dart - free$ *graphs*
(b) 1) $\nexists n \in N$, $\nexists P_3 : abc$, $P_3 \in [A \cup R]$ *such that* $\{n\} \sim \{a,b,c\}$;
2) $\nexists v \in A$, $\nexists P_3 : xyz$, $P_3 \in [N]$ *such that* $\{v\} \sim \{y\}$ *and* $\{v\} \not\sim \{x,y\}$;
(c) $\forall e = ab \in [N]$ *with* $(N(a) \cap N(b) \cap A \neq \emptyset$ *or* $N(e) \cap R$ *is not clique) implies* $(N(a) \cap A) - N(b) = \emptyset$ *and* $(N(b) \cap A) - N(a) = \emptyset$.

Proof. Let *G* be $dart$-free. Because the property of being $dart$- free is hereditary it results $G - R$ and $G - A$ are $dart$-free. If it would exist $n \in N$ and $P_3 : abc \in [A]$ such as $\{n\} \sim \{a,b,c\}$ then $[\{a,b,c,n,r\}]$ is dart, for every $r \in R$, because $R \sim N$. Supposing that it would exist $n \in N$ and $P_3 : abc$, $P_3 \in [R]$ such as $\{n\} \sim \{a,b,c\}$ then $[\{a,b,c,n,t\}]$ is dart, because N is the neighborhood of A and consequently there exists $t \in A$, $\{t\} \sim \{n\}$.

Supposing that there exists $v \in A$ and $P_3 : abc$, $P_3 \in [N]$ such as $vb \in E$ and $va, vc \notin E$ then $[\{a,b,c,v,r\}]$ is dart, for every $r \in R$, because $R \sim N$. Suppose that there exists $e = ab \in [N]$ such as $N(a) \cap N(b) \cap A \neq \emptyset$ and we still have $(N(a) \cap A) - N(b) \neq \emptyset$. Let $x \in N(a) \cap A \cap N(b)$ and $y \in (N(a) \cap A) - N(b)$. Because $R \neq \emptyset$ and $R \sim N$ it follows that $\{a,b\} \sim \{r\}$, $\forall r \in R$, which means that $[\{a,b,x,y,r\}]$ is dart. Suppose that there exists $e = ab \in [N]$ such as $N(e) \cap R$ is not a clique and still we have $(N(a) \cap A) - N(b) \neq \emptyset$. Let $x \in (N(a) \cap A) - N(b)$. Because $N(e) \cap R$ is not a clique, $R \neq \emptyset$ and $R \sim N$ it follows that there exists $u, v \in R$ with $\{a,b\} \sim \{u,v\}$ and $uv \notin E$, which means that $[\{a,b,u,v,x\}]$ is dart.

Suppose that conditions a), b) and c) are satisfied and still exists $[D] = [\{l,m,n,p,q\}]$ dart, induced subgraph of G such as $d_{[D]}(q) = 1$, $d_{[D]}(l) = d_{[D]}(n) = 2$, $d_{[D]}(m) = 3$ and $d_{[D]}(p) = 4$. Because $G - A$ and $G - R$ are dart-free graphs, G and $[D]$ are convex graphs, $A \not\sim R$ it folows that $D \cap A \neq \emptyset$, $D \cap N \neq \emptyset$, $D \cap R \neq \emptyset$. Because $\{p\} \sim D - \{p\}$, $A \not\sim R$, $D \cap A \neq \emptyset$, $D \cap R \neq \emptyset$ it follows that $p \notin A \cup R$, which means that $p \in N$. According to (b) 1), because $p \in N$, $\{p\} \sim \{l,m,n\}$ it follows that $\{l,m,n\} \not\subset A$. (denoted 3)). According to (b) 1), because $p \in N$, $\{p\} \sim \{l,m,n\}$ it follows that $\{l,m,n\} \not\subset R$. (denoted 4)).

i) If $m \in A$, as $A \not\sim R$ and $\{m\} \sim \{l,n,p\}$ it follows that $\{l,n\} \subseteq A \cup N$. From (3) it results that $\{l,n\} \not\subset A$. Suppose $l \in N$. Because $R \sim N$ and $ql \notin E$ it follows that $q \notin R$. Vertex $q \notin A$ (otherwise: for $n \in R$, from $R \sim N$, results $nl \in E$, a contradiction; for $n \in A \cup N$, results $D \cap R = \emptyset$, a contradiction). Supposing that $q \in N$, because $D \cap R \neq \emptyset$ it follows that $n \in R$, but, in this case, from $R \sim N$, results $qn \in E$, a contradiction. Consequently, $q \notin N$. So $l \notin N$. Similarly, it can be proved that $n \notin N$. Because $\{l,n\} \subseteq A \cup N$ it follows that $\{l,n\} \subseteq A$, which contradicts the fact that $\{l,m,n\} \not\subset A$. So, $m \notin A$.

ii) If $m \in R$, because $A \not\sim R$ and $\{m\} \sim \{l,n,p\}$ it follows that $\{l,n\} \subseteq R \cup N$. From 4) results $\{l,n\} \not\subset R$. Suppose $l \in N$. Because $R \sim N$ and $ql \notin E$ it follows that $q \notin R$. The vertex $q \notin R$ (otherwise, from $R \sim N$, results $mq \in E$, a contradiction). Supposing that $q \in A$, as $mn \in E$ and $A \not\sim R$ it follows that $n \in N \cup R$. Because $ln \notin E$ and $R \sim N$, it results that $n \notin R$. So $n \in N$. But, in this case, (b) 2) is contradicted. Consequently, $q \notin A$. So $l \notin N$. Similarly, it can be prooved that $n \notin N$. Because $\{l,n\} \subseteq R \cup N$ it follows that $\{l,n\} \subseteq R$, contradicting the fact that $\{l,m,n\} \not\subset R$. So, $m \notin R$.

iii) Suppose $m \in N$. If $q \in R$ then $mq \in E$, a contradiction. If $q \in N$, because $D \cap R \neq \emptyset$ it follows that l or n belong to R and then $lq \in E$ or $nq \in E$, a contradiction. So $q \in A$. Because $D \cap R \neq \emptyset$ it results that l or n belong to R. If $\{l,n\} \subseteq R$, as $R \sim N$ and $ln \notin E$ results that $(N(p) \cap A) - N(m) = \emptyset$, a contradiction. Consequently $\{l,n\} \not\subset R$. Suppose that $l \in R$. Then $n \notin R$. As $\{l,n\} \not\subset R$, $ln \notin E$ and $R \sim N$ results that $n \notin N$. It follows that $n \in A$. We have $mp \in E([N])$ with $N(m) \cap N(p) \cap A \neq \emptyset$ and still $(N(p) \cap A) - N(m) \neq \emptyset$, contradicting c).

The above result leads to the following recognition algorithm.

Input: $G = (V,E)$ a connected graph with at least two nonadjacent vertices.
Output: An answer to the question: Is G a *dart*-free graph ?

1. { $L = G$; // L a list of graphs
2. *while* ($L \neq \emptyset$)
   { extract an element H from L;
   find a weakly decomposition (A,N,R) for H;
   if ($\exists n \in N, \exists P_3 : abc, P_3 \in [A \cup R]$ *such that* $\{n\} \sim \{a,b,c\}$)
   or ($\exists v \in A, \exists P_3 : xyz, P_3 \in [N]$ *such that* $\{v\} \sim \{y\}$
   and $\{v\} \not\sim \{x,y\}$)
   or (($\exists e = ab \in E([N])$ *such that* ($N(a) \cap N(b) \cap A \neq \emptyset$
   or $N(e) \cap R$ *is not clique*)
   and (($N(a) \cap A) - N(b) \neq \emptyset$ *or* $(N(b) \cap A) - N(a) \neq \emptyset$)) *then*
   Return: G isn't $dart - free$;
   else introduce in L the connected components of $G - R$, $G - A$ incomplete
3. *Return:* G is $dart - free$ }
   }

References

[1] L. Babel, A. Brandstadt, V. B. Le, "Recognizing the P_4-structure of claw-free graphs and a larger graph class", *Discrete Mathematics and Theoretical Computer Science* 5, 2002, 127-146.

[2] V. Chvatal, J. Fonlupt, L. Sun, A. Zemirline, "Recognizing Dart-Free Perfect Graphs", *SIAM JOURNALS ONLINE, SICOMP*, Volume 31 Issue 5, pages 1315-1338, 2002.

[3] C. Croitoru, M. Talmaciu, "A new graph search algorithm and some applications", presented at *ROSYCS 2000*, Univ. "Al. I. Cuza" Iaşi, 2000.

[4] C. Linhares Sales, F. Maffray, "On dart-free perfectly contractile graphs", *Theoretical Computer Science*, Volume 321, Issues 2-3, 2004, pg. 171-194.

[5] M. Talmaciu, *Decomposition problems in graph theory, with applications in combinatorial optimization* , PhD Thesis, Univ. "Al. I. Cuza" Iasi, 2002.

Nechita Elena, Talmaciu Mihai, Crisan Gloria Cerasela
University of Bacău
Department of Mathematics and Informatics
Address: 8 Spiru Haret St., 600114 Bacău, Romania
E-mail: {elenechita,mihaitalmaciu,ceraselacrisan}@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 336-340

Performance Analysis of Spatial Data Indexing

Bogdan Oancea, Razvan Zota

Abstract: Applications likes spatial databases systems, CAD, GIS, robotics have to be able to manage large volumes of data and require for access methods to perform spatial queries in reasonable time. The most used access method for spatial data is the R-tree or one of its multiple versions. In this paper we present the main characteristics of R-trees and R*-trees and propose an improved insertion algorithm. Experiments showed that our proposal improved the speed of insertion operation into an R-tree with about 30 percent comparing with the original R*-tree.

Keywords: spatial access methods, R-tree, spatial data indexing

1 Introduction

Spatial data arise in several applications, including geographical information systems (GIS), computer-aided design (CAD), computer vision and robotics. Therefore, spatial database systems and other applications designed to store, and operate spatial data have received considerable attention over the last years. These systems have to be able to manage large volumes of data and require for access methods to perform spatial queries in reasonable time. Disk based index structures for spatial data have been researched extensively - see for example the survey by Gaede and Gaunther [7].

Especially the R-tree [3] and its numerous variants have emerged as practically efficient indexing methods. R-tree applications cover a very wide spectrum, from spatial and temporal to multimedia databases. The initial application that motivated Guttman to his research was VLSI design - how to efficiently answer whether a space is already covered by a chip or not. Handling rectangles quickly found application in geographical and spatial data, including GIS image or video/audio retrieval systems, time series and chronological databases.

Nowadays, spatial databases and geographical information systems have been established as a mature field, spatiotemporal databases and manipulation of moving points and trajectories are being studied extensively, and multimedia databases able to handle new kinds of data like images, music, or video, are being developed. An application in all these cases should rely to R-trees as a necessary tool for data storage and retrieval.

This paper is organized as follows: in section 2 we make an overview of the original R-tree and the R*-tree that proves to be the most efficient and widely-used variant of R-trees. In section 3 we propose a new strategy for treatment of the overflow, section 4 presents experimental results while section 5 concludes the paper.

2 R-tree

R-trees are hierarchical data structures based on the B+trees. They are used for the dynamic organization of a set of d-dimensional geometric objects representing them by the minimum bounding d-dimensional rectangles (called MBR in the rest of the paper). Each node of the R-tree corresponds to the MBR that bounds its children. The leaves of the tree contain pointers to the database objects instead of pointers to children nodes.

It must be noted that the MBRs that surround different nodes may overlap each other which means that a spatial search may visit many nodes before confirming the existence of a given MBR. Also, it is easy to see that the representation of geometric objects through their MBRs may result in false alarms. To resolve false alarms, the candidate objects must be examined.

An R-tree of order (m, M) has can be defined as follows [3]:

- 1. Unless it is the root, each leaf node can accomodate up to M entries, whereas the minimum allowed number of entries is $m \geq M/2$. Each entry is of the form (MBR, ObjID), such that MBR is the minimum bounding rectangle that spatially contains the object and ObjID is the objects identifier;
- 2. An internal node can store a number of entries between $m \geq M/2$ and M. Each entry is of the form (MBR, ptr), where ptr is a pointer to a child of the node and MBR is the minimum bounding rectangle that spatially contains the MBRs contained in this child;
- 3. The minimum allowed number of entries in the root node is 2, unless it is a leaf;

- 4. All leaves of the R-tree are at the same level;

From the above definition of the R-tree it is simple to conclude that it is a height-balanced tree. Figure 1 shows the MBRs of some geometric data and figure 2 shows a way a organizing these rectangles in an R-tree (it is obvious that there are several R-trees that can represent the MBRs, depending on the order of inserstions).

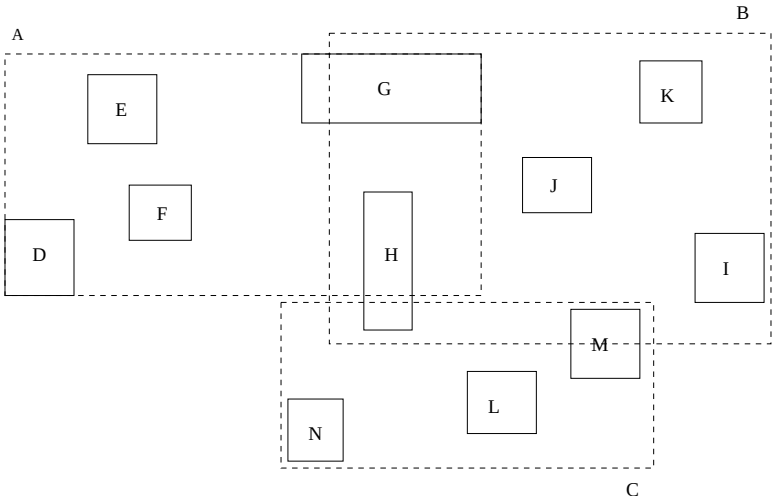


Figure 1: An example of some MBRs

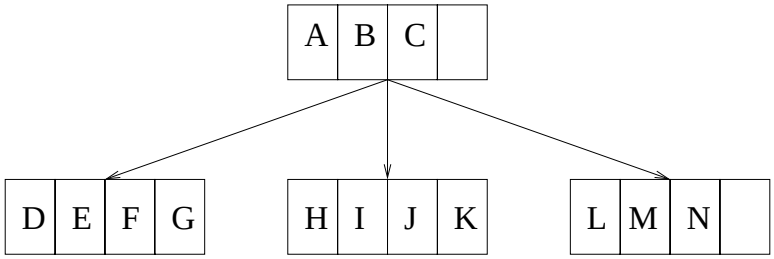


Figure 2: The corresponding R-tree

Insertions of new objects into an R-tree are directed to leaf nodes. The node that will be least enlarged is chosen at each level. Finally, the object is inserted in an existing leaf if there is space for a new entry, otherwise a split takes place. Guttman, in the original paper [3] proposed three alternative algorithms to handle splits, which are of exponential, quadratic and linear complexity:

- Exponential: all possible groupings are exhaustively tested and the best is chosen with respect to the minimization of the MBR enlargement.
- Quadratic: choose two objects as seeds for the two nodes. These objects if put together should create as much dead space as possible (dead space is the space that remains from the MBR if the areas of the two objects are ignored). Until there are no remaining objects, choose for insertion the object for which the difference of dead space if assigned to each of the two nodes is maximized, and insert it in the node that requires smaller enlargement of its respective MBR.
- Linear split: choose two objects as seeds for the two nodes, where these objects are as furthest as possible. Consider each remaining object in a random order and assign it to the node requiring the smaller enlargement of its respective MBR.

Guttman proposed using the quadratic algorithm as a good compromise in order to achieve a reasonable performance for spatial queries. For a detailed description of the algorithms used for inserting, deleting a node into n R-tree or searching an object in an R-tree the reader is invited to consult the original paper [3].

Proposed in 1990 [4], R*-trees are widely accepted in the literature as a prevailing performance-wise structure that is often used as a basis for performance comparisons. The R*-tree follows a sophisticated node split technique called forced reinsertion. According to this technique, when a node overflows, p entries are extracted and reinserted in the tree (p being a parameter, with 30 percent a suggested optimal value).

The R-tree is based solely on the area minimization of each MBR. On the other hand, the R*-tree goes further considering the following criteria:

- Minimization of the area covered by each MBR. This criterion minimize the dead space to reduce the number of paths pursued during query processing.
- Minimization of overlap between MBRs. The larger the overlapping, the larger is the expected number of paths followed for a query. This criterion has the same objective as the previous one, i.e. reducing the number of paths pursued during query processing.
- Minimization of MBR perimeters. This criterion tries to give more quadratic rectangles, to improve the performance of queries that have a large quadratic shape.
- Maximization of storage utilization. When utilization is low, more nodes tend to be invoked during query processing.

The R*-tree follows an engineering approach to find the best possible combinations of these criteria. This approach is necessary, because the criteria can become contradictory.

For the insertion of a new entry, one have to decide which branch to follow, at each level of the tree. This algorithm is called ChooseSubtree. The complete description of this algorithm can be found in [4].

In case ChooseSubtree selects a leaf that cannot accommodate the new entry the R*-tree does not immediately resort to node splitting. Instead, it finds a fraction of the entries from the overflowed node and reinserts them. The set of entries to be reinserted are those whose centroid distances from node centroid are among the largest 30 percent. The reinsertion algorithm achieves a tree rebalancing and improves performance during query processing. However, reinsertion is a costly operation. Therefore, only one application of reinsertion is permitted for each level of the tree. When overflow cannot be handled by reinsertion, node splitting is performed according to a elaborate algorithm.

3 A strategy for the treatment of the overflow

Our experimental studies showed that forced reinsertion is a time consuming technique that makes the building procedure of an R*-tree by repeated insertions of the new nodes time critical. Instead of this classical approach we propose a technique that significantly reduces the CPU time needed to build the R-tree by insertions.

Our insertion heuristic improves the shape of the R-tree so that the tree achieves a more efficient shape, the number of nodes being smaller and with an improved storage utilization. This technique redistributes data among neighboring nodes when it is possible in order to reduce the total number of newly created nodes.

In the case of node overflow, we can examine all other nodes to see if there is another node at the same level able to accommodate one of the overflowed nodes rectangles. Thus, the split could be prevented. A split is performed only when all nodes are completely full. To reduce the insertion cost, we have to redistribute an entry from the overflowed node we do not examine all nodes but only the neighboring nodes when.

In case of node overflow, our algorithm proceeds as follows :

- 1. examines the parent node, to find the MBRs that overlap the MBR of the overflowed node.
- 2. each record in the overflowed node is examined to see if it can be moved to the nodes corresponding to the previously found overlapping MBRs.
- 3. records are moved only if the resulting area of coverage for the involved nodes does not have to be increased after the moving of records.
- 4. if no movement is possible, a normal node split takes place.

This approach will reduce the time needed for insertion, fact that is proved by our experiments. The time for a spatial query is about the same like R*-trees but the memory usage is improved about 95 percent of nodes are full in out approach, comparatively with R*-trees that have a fill factor of about 70 percent.

4 Experimental results

We have conducted a series of experiments to show that the algorithm described above is faster than the forced reinsert used by the original R*-trees. We have implemented the original version of the R*-tree and the modified one using the Java programming language.

Then, we have measured the time needed for building the tree in these two versions using 3 synthetic data sets: the first data set consists of 100.000 polygons, the second has 200.000 polygons and the third data set has 400.000 polygons. All polygons are uniform distributed in the square [(0; 1) ,(0;1)]. After building the tree we executed a number of spatial queries, where the query window has a square shape and is uniform distributed over the MBR of the data set.

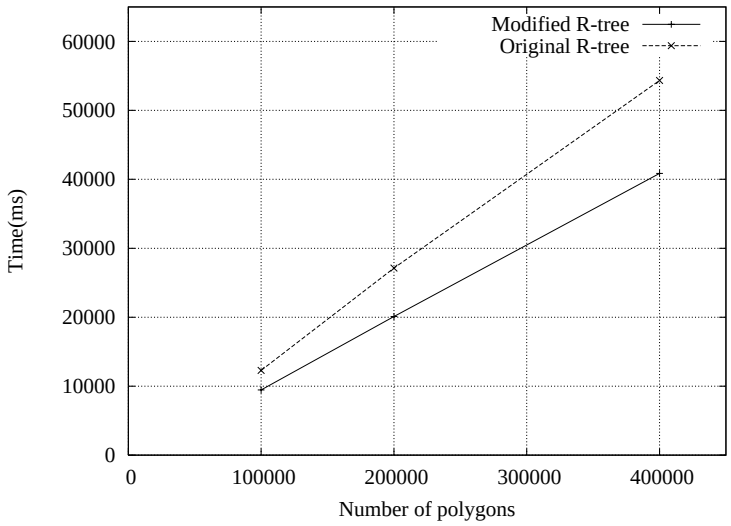


Figure 3: Time needed for building the tree

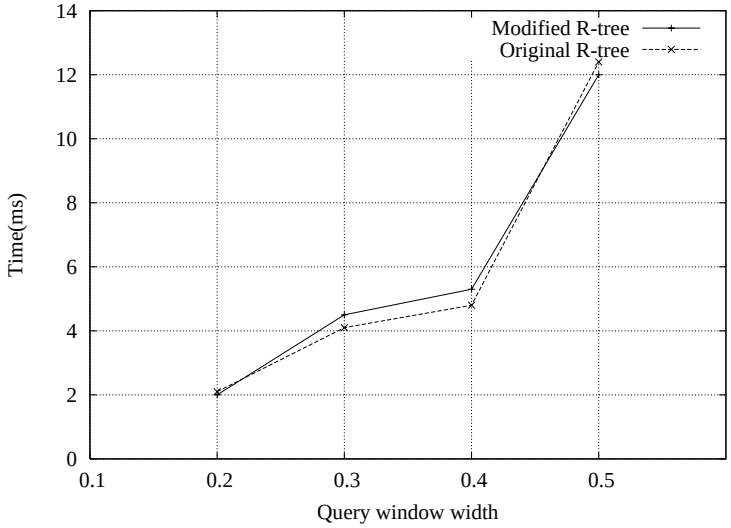


Figure 4: The average time needed for a spatial query

The programs was executed on a computer with an AMD Duron processor at 1.4GHz and 512 MB of main memory. The Java virtual machine was started with -Xmx400Mb option. Figures 3 and 4 shows the results.

From these figures in can easily be concluded that our modified version of the R-tree is about 30 percent faster then the R*-tree regarding the building of the tree by repeated insertion. These results is due to the fact that our

modified version reduces the number of created nodes, avoiding splitting of a node until it is absolutely necessary.

Thus, our locality approach makes the procedure of dynamically building the tree faster and also using less memory than the original version of the R*-tree. The spatial query time is almost the same in the two cases: our modified R-tree and the original R*-tree.

5 Summary and Conclusions

Since Guttman [3] first proposed the R-tree in 1984 as an efficient method for spatial data indexing, many applications like computer vision, CAD, GIS, multimedia databases uses a variant of the R-tree as a spatial access method. R*-trees, a version of R-tree, are widely accepted in the literature as a prevailing performance-wise structure that is often used as a basis for performance comparisons. In this paper we compared the performance of the insertion algorithm of the R*-tree with our version that replaces the forced-reinsertion technique with a new one described above. Our experiments showed that our approach improves the speed of insertion operation with about 30 percent comparing with the original R*-tree approach.

References

[1] A. Abulnaga and J.F. Naughton, “Accurate Estimation of the Cost of Spatial Selections”, *Proceedings 16th IEEE ICDE Conference*, pp.123-134, San Diego, CA, 2000.

[2] P.K. Agarwal, M. deBerg, J. Gudmundsson, M. Hammar and H.J. Haverkort, “Box-trees and R-trees with Near Optimal Query Time”, Proceedings Symposium on Computational Geometry, pp.124-133, Medford, MA, 2001.

[3] A. Guttman, “R-trees: a Dynamic Index Structure for Spatial Searching”, Proceedings ACM SIGMOD Conference, pp.47-57, Boston, MA, 1984.

[4] N. Beckmann, H.P. Kriegel, R. Schneider and B. Seeger, “The R*-tree: an Effcient and Robust Method for Points and Rectangles”, Proceedings ACM SIGMOD Conference, pp.322-331, Atlantic City, NJ, 1990.

[5] S. Brakatsoulas, D. Pfoser and Y. Theodoridis, “Revisiting R-tree Construction Principles”, Proceedings 6th ADBIS Conference, pp.149-162, Bratislava, Slovakia, 2002.

[6] Y. Garcia, M. Lopez and S. Leutenegger, “On Optimal Node Splitting for R-trees”, Proceedings 24th VLDB Conference, pp.334-344, New York, NY, 1998.

[7] V. Gaede and O. Guenther, “Multidimensional Access Methods”, ACM Computing Surveys, Vol.30, No.2, pp.170-231, 1998.

[8] Y. MANOLOPOULOS, A. NANOPOULOS, A. N. PAPADOPOULOS, “R-trees Have Grown Everywhere”, ACM Computing Surveys, Vol. V, No. N, Month 20YY.

Bogdan Oancea
Artifex University
Address: 47, Econom Cezarescu St., sector 5, Bucharest
E-mail: bogdan.oancea@titan-software.com

Razvan Daniel Zota
ASE Bucharest
Address: 6, Piata Romana, sector 1, Bucharest
E-mail: zota@ase.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 341-348

Mining Multi-Level Association Rules Using FP-Tree and AFOP-Tree

Mirela Pater, Cornelia Győrödi, Robert Győrödi, Alina Bogan-Marta

Abstract: Association rule mining is a central problem in discovering knowledge. It finds interesting association or correlation relationships among a large set of data items. In this paper, the horizon of frequent pattern mining is expanded by extending single-level algorithms for mining multi–level and multi–dimensional frequent patterns. There are presented two algorithms that extract multi-level association rules from databases using two efficient data structures: FP-Tree and AFOP-Tree. A comparison study is made between these algorithms. The compared algorithms are presented together with some experimental data that leads to the final conclusions.

1 Introduction

The explosive growth of many business and scientific databases has far outpaced our ability to interpret and digest this data. Data Mining refers to extracting or "mining' knowledge from large amount of data. Data mining therefore appears as a useful tool to address the need for sifting useful information such as hidden patterns from databases. The aim of data mining is the discovery of patterns within data stored in databases.

Association mining searches for interesting relationship among items in a given database and displays it in a rule form, i.e. as "if customers buy product x then they also buy product y" [4]. With the massive amounts of data continuously being collected and stored in databases, many industries are becoming interested in mining associations among data. Market basket analysis is a typical example among various applications of association mining. The association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold [3].

Finding frequent itemsets is one of the most investigated fields of data mining. The problem was first presented by Agrawal [2]. The subsequent paper of Agrawal and Srikant [1] is considered as one of the most important contributions to the subject. Its main algorithm, APRIORI, not only influenced the association rule mining community, but it affected other data mining fields as well. Association rule and frequent itemset mining became a widely researched area, and hence faster and faster algorithms have been presented.

Frequent pattern mining plays an essential role in mining associations, as shows Agrawal and Srikant [2] and Klemettinen et al. [7], in multi-level and multi-dimensional patterns, as shows Runying Mao [6], and many other important data mining tasks. Mining frequent patterns in transaction databases, multi-level databases, and many other kinds of databases has been studied popularly in data mining research.

A very influential association rule mining algorithm, APRIORI [1], has been developed for rule mining in large transaction databases. Many other algorithms developed are derivative and/or extensions of this algorithm. A large step forward in improving the performances of these algorithms was made by introduction of a novel, compact data structure, called frequent pattern tree, or FP-TREE [2], and the associated mining algorithms, FP-GROWTH [3]. Although these studies are on sequential data mining techniques, algorithms for multi-level mining of association rules have also been proposed [5],[6][8],[9]. Later, another compact data structure was proposed to represent conditional databases, called Ascending Frequency Ordered Prefix-Tree – AFOPT [10] more efficient.

Most of the previous studies, adopt a heuristic/greedy search techniques, APRIORI -like approach, which is based on an anti-monotone Apriori heuristic described by Agrawal and Srikant [1]: if any length k pattern is not frequent in the database, its length (k +1) super-pattern can never be frequent.

The essential idea is to iteratively generate the set of candidate patterns of length (k + 1) from the set of frequent patterns of length k (for $k \geq 1$), and check their corresponding occurrence frequencies in the database.

This paper presents and compares two algorithms implemented using this two data structure (FP-Tree and AFOP-Tree) for representing conditional databases.

2 Multi-level frequent pattern mining

For mining multiple-level association rules, concept taxonomy should be provided for generalizing primitive level concepts to high level ones.

Frequent pattern mining is used in many data mining applications, e.g., market-basket data analysis, web log mining and biological data mining. It can also serve as a feature selection tool for classification and clustering. The problem was first introduced by Agrawal [1] in the context of transactional databases.

It can be stated as follows: Let I = { i_1, i_2, . . ., i_n } be a set of items and D = { t_1, t_2, . . . , t_N}be a transaction database, where t_i (i ∈ [1, N]) is a transaction and $t_i \subseteq$ I. Every subset of I is called an *itemset*. If an itemset contains k items, then it is called a k-itemset. The support of an itemset l in D is defined as the percentage of transactions in D containing l, i.e. support(l) $= \|\{t | t \in D \bigwedge I \subseteq t\}\| / \|D\|$. If the support of an itemset exceeds a user-specified minimum support threshold, then the itemset is called a frequent. The task of the frequent pattern mining problem is given a transaction database and a minimum support threshold, to enumerate all the frequent patterns.

The main issues in frequent patterns mining are: (1) to reduce the database scanning times since in many cases the transactional database is too large to fit into the main memory, and scanning data from disk is very costly; (2) to reduce the search space since every subset of I can be frequent and the number of them is exponential to the number of items in I ; and (3) to count support efficiently, naive subset matching is quite costly due to the large size of the database and the large number of potential frequent itemsets.

Multi-level frequent pattern mining is a very promising research topic and plays an invaluable role in real life applications. The classic frequent pattern mining algorithms (APRIORI, FP-Growth) have been focusing on mining knowledge at single concept levels. It is often desirable to discover knowledge at multiple concept levels that are interesting and useful.

To find relatively frequently occurring patterns and reasonably strong rule implications, a user or an expert may specify two thresholds: minimum support, and minimum confidence. Notice that for finding multiple-level association rules, different minimum support and/or minimum confidence can be specified at different levels. The process of mining association rules is expected to first discover large patterns and strong association rules at the top-most concept level.

A method for mining multiple-level association rules uses a hierarchy information encoded transaction table, instead of the original transaction table, in iterative data mining.

3 Multi-level FP-Growth algorithm (MLFP-Growth)

The Apriori heuristic achieves good performance gain by (possibly significantly) reducing the size of candidate sets. However, in situations with prolific frequent patterns, long patterns, or quite low minimum support thresholds, an Apriori -like algorithm may still suffer because it is costly to handle a huge number of candidate sets and it is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

FP-Tree's efficiency of mining is achieved with three techniques: (1) a large database is compressed into a highly condensed, much smaller data structure, which avoids costly, repeated database scans, (2) FP-tree-based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets, and (3) a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined patterns in conditional databases, which dramatically reduces the search space.

The FP-growth method is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm and also faster than some recently reported new frequent pattern mining methods. FP-growth is at least an order of magnitude faster than Apriori, and such a margin grows even wider when the frequent patterns grow longer. Many strong associations discovered at rather high concept levels are common sense knowledge. Therefore, a mining system with the capabilities to mine association rules at multiple levels of abstraction and traverse easily among different abstraction spaces is more desirable.

Multi-level databases use hierarchy-information encoded transaction table, in the transaction table each item is encoded as a sequence of digits [12].

Example: The item "Fruit' is encoded as 0.1 according to the example below (Fig. 1). The first digit '0' represents "Root' at the first level, the 2^{nd} digit represents "Fruit' in the 2^{nd} level, 3^{rd} level the first digit "0.1.1' is "Apple". Digit's can be used to transform items into categories. Sometimes the information is not straightforward but it can be obtained easily. The information required to create such a database is either implicit (for example the cabbage is a vegetable which can be considered food) or it's provided by the user.

Table 1, contains category codes and a description for each code (category or item) which is only needed for the final display. The codes and the descriptions for table 1 are extracted from figure 1.

Let I = { a_1,... a_m} be a set of items, and a transaction database DB = {T_1, T_2,... T_n}, where T_i (i∈[1..n]) is a transaction which contains a set of items in I.

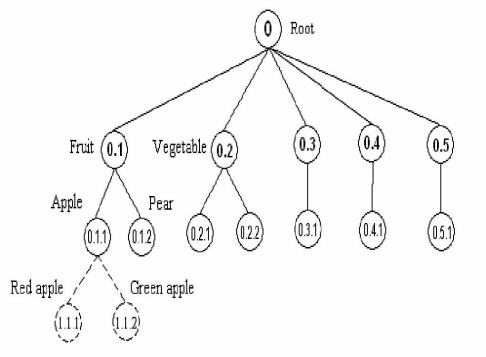


Figure 1: Hierarchy-base multi-level database

Code	Description
0	Root
01	Fruit
02	Vegetable
011	Apple
012	Pear
0111	Red Apple
0112	Green Apple

Table 1: Category code and description

Mining single-level databases, is like mining information from the lowest concept-level, which contains the products IDs (like 0.1.1.1, or milk 2% in real life). This information is sometimes too dense, and unnecessary, because more general information is required.

By using the FP-Tree algorithm only 2 scans of the database are made and no candidate sets are being generated, thus the top-down progressive method doesn't help. The information gained at a high concept level is insufficient for the following concept levels. Thus, a new scan of the database is required to get the missing information. This way, to get information from the lowest concept level will require at least n+1 scans of the database.

In conclusion, this method is inefficient when using FP-Tree. Still, when getting information from the lowest level, the FP-Growth algorithm, directly used on the last concept level, is faster than multi-level Apriori [15]. To get the information needed for the k concept level there are 2 methods:

a. directly use FP-Tree on the requested concept level

b. if information from a level lower than k is available , information for the K level can be obtained from the L-level (L>K) (FP-Tree without scanning the database, or by scanning just a few transactions

3.1 FP-Tree directly on the K-concept level

We consider a database with 3 concept levels which has items like the ones in figure 1 (the root is not considered a concept level in this example). The database may have many more concept levels, however only 3 are used for the examples from sections 5 and 6.

Using directly the FP-Growth algorithm on the lowest concept level (or the level we need) will require only 2 scans of the database as opposed to k+1 scans with the method described at 3. We consider a database with the transactions presented in the left part of table 1. It can be seen that each item is identified by 3 numbers. Each number identifies the item on its respective concept level.

If we are interested in 2^{nd} level information (frequent itemsets from the 2^{nd} concept level) we will use only the first 2 numbers from the item identifier. If the minimum support is 3 then, after the first step of the FP-Tree algorithm we get the results presented in the right part of table 2. During the 2^{nd} step of the algorithm we get the following tree (figure 2) from which the information requested can be extracted using the FP-Growth algorithm.

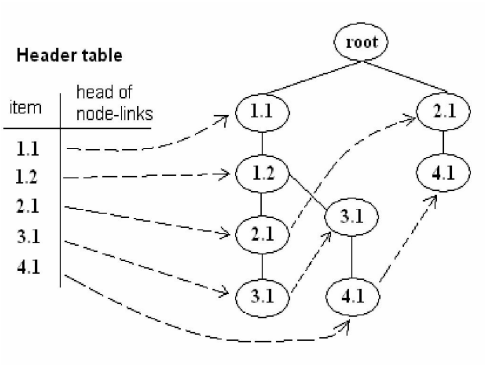


Figure 2: 2^{nd} concept level FP-Tree

TID	Items Bought	(Ordered)Frequent 2^{nd}-lvl categories
1	3.1.1, 1.2.2, 2.2.3, 2.1.1, 1.1.4	1.1, 1.2, 2.1, 3.1
2	2.1.2, 4.1.1	2.1, 4.1
3	2.1.2, 4.1.2, 5.1.3	2.1, 4.1
4	1.2.3, 1.1.2, 3.1.1, 4.1.1	1.1, 1.2, 3.1, 4.1
5	1.1.2, 3.1.1, 1.2.3	1.1, 1.2, 3.1

Table 2: A transaction database as running example

Notice that items (categories) 2.2 and 5.1 have been pruned from the header because they do not meet minimum support.

3.2 Obtaining a K-level FP-Tree from a L-level one

This method allows us to obtain information for level K when have already obtained the results for level L which is a lower concept level than K. The FP-Tree obtained for level L contains almost information to build the FP-Tree required for level K the only information missing is the one from the items that did not meet minimum support. These items may affect the K-level FP-Tree.

In step 1, the new K-level FP-Tree needs a header which has k-level items. This is header is partial and may be completed at step 2. Step 3 complete the L-level tree and add to the tree the part of the transactions which contain the added nodes. Then, in step 4, the K-level tree header was completed at step 2 and contains item IDs corresponding to concept level K. They might need to be reordered by support. To find the support of items from level K we use a recursive algorithm on the L-level FP-Tree. Finally, in step 5, starting from each terminal item of the L-level tree we follow their conditional patterns and extract all the K-level items that appear and their support. If a k-item appears more than once we consider it only once but sum their supports. We create an itemset which is added to the new K-level tree just like a transaction would be added to an FP-Tree. In the end a K-level FP-Tree is obtained (figure 3), from which, by using the FP-Growth algorithm, we can obtain the frequent itemsets from the first concept level.

Notice that category 5 was pruned because it didn't meet minimum support. Although it is rare in practice, if no items were pruned from the L-level FP-Tree, then steps 2 & 3 are not required, and the database is not accessed at all. Thus, if we have the FP-Tree for the lowest concept level with no pruned items, then we can obtain information for any concept level without accessing the database. This is useful to know when we need information for more than one concept level

4 Multi-level AFOPT algorithm (ML-AFOPT)

The algorithm is obtained by extending the classic AFOPT algorithm [9] for multi-level databases ML-AFOPT [12]. The algorithm uses a compact data structure to represent the conditional databases, and the tree is traversed

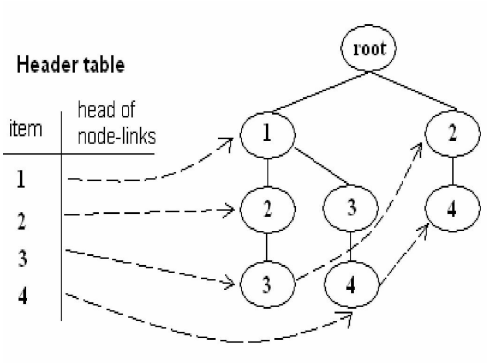


Figure 3: K-level FP-Tree

top-down. The combination of the top-down traversal strategy and ascending frequency order minimizes both the total number of conditional databases and the traversal cost of individual conditional databases. By using the AFOPT algorithm no candidate sets are being generated, thus the top-down progressive method doesn't really help. The information gained at a high concept level is insufficient for the following concept levels. Thus, a new scan of the database is required to get the missing information. This way, to get information from the lowest concept level will require at least n+1 scans of the database. In conclusion, this method is inefficient when using pattern algorithms. Still, when getting information from the lowest level, the AFOPT algorithm, directly used on the last concept level, is faster than candidate generation and testing algorithms for multi-level.

The AFOPT structure is a compact representation of the conditional databases. It contains the complete information for mining frequent itemsets from the original database. The size of the AFOPT structure is bounded by, but usually much smaller than, the total number of frequent item occurrences in the database.

Like in AFOPT, this algorithm, ML-AFOPT [12], first traverses the original database to find frequent items or frequent abstract levels. These are sorted in ascending order and sort them in ascending frequency order. Then the original database is scanned the second time to construct an AFOPT structure to represent the conditional databases of the frequent items. First, a header is constructed, using only the items (categories) that pass the minimum support threshold, and which are ordered in an ascending order, unlike in the FP-Growth algorithm which items are in descending frequency order.

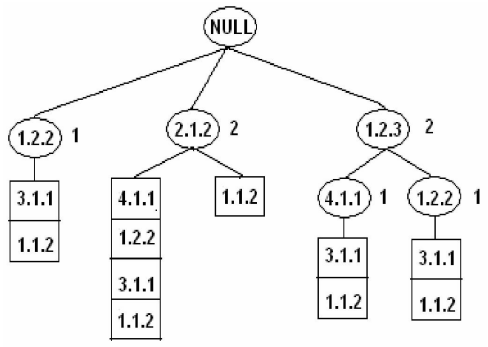


Figure 4: AFOP-Tree

The conditional database of an item i include all the transactions containing item i, and infrequent items and those items before i are removed from each transaction. Arrays are used to store single branches in the AFOPT structure to save space and construction cost. Each node in the AFOPT structure contains three pieces of information: an item id, the count of the itemset corresponding to the path from the root to the node, and the pointers pointing to the children of the node.

To get the information needed for the k concept level there are 2 methods, like in MLFP-Growth algorithm:

a. directly use AFOPT on the requested concept level

Item	Sup
3.1.1	4
1.2.2	3
2.2.3	1
2.1.1.	1
1.1.4	1
1.1.2	5
2.1.2	2
4.1.1	2
4.1.2.	1
5.1.3	1
1.2.3	2

Table 3: Items from database

b. if information from a level lower than k is available , information for the K level can be obtained from the L-level (L

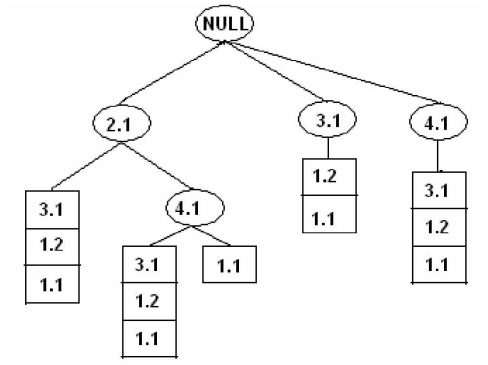


Figure 5: K-level AFOP-Tree

This method allows us to obtain information for level K when we have already obtained the results for level L which is a lower concept level than K. In the end a K-level AFOP-Tree is obtained (figure 5), from which, by using the FP-Growth algorithm, we can obtain the frequent itemsets from the first concept level.

5 Comparison study and conclusions

Using multiple-level databases we have the ability to focus our attention on discovering informative rules between categories, or even between an item and a category, not just rules between items. Using candidate set generation is costly, especially when there are prolific patterns and/or long patterns.

Given a conditional database, two different structures are created to represent it a particular traversal strategy for each structure is used: (1) the FP-tree structure (items are ordered in descending frequency order) and the bottom-up traversal strategy, and (2) the AFOPT (items are ordered in ascending frequency order) structure and the top-down traversal strategy.

If a conditional database can be represented by a single branch using FPtree structure, then it can also be represented by a single branch using the AFOPT structure, however if a conditional database can be represented by a single branch in the AFOPT structure, it may contain multiple branches in the FP-tree. If the cost of visiting a single node in the above structures is assumed the same, then, the AFOPT structure combined with the top-down traversal strategy needs the least traversal cost. Thus, using the AFOPT structure, the number of node visits is minimal.

One of the important factors that slow the FP-Tree algorithm is database access. The more distance there is between concept levels K and L the more missing information there may be which forces a costly database access.

If no information is mission between levels K and L the only reason to access the database would be for support check and sometimes not even that is required.

Descending frequency order is used in FP-tree. This improves the possibility of prefix sharing. Hence FP-tree is more compact than the AFOPT structure. It is possible that all the conditional databases of a frequent itemset's frequent extensions can be represented by a single branch using FP-tree structure, and the AFOPT structure requires multiple branches. In this case, both algorithms do not need to construct new conditional databases, but the AFOPT algorithm needs more traversal cost. The possibility that this situation happens is much lower than the possibility that a single conditional database can be represented by a single branch.

The AFOPT structure may contain more nodes than the FP-tree structure because the ascending frequency order reduces the possibility of prefix sharing. However, the FP-growth algorithm needs to maintain parent links and node-links at each node, which incurs additional construction cost and consumes more space. Arrays are used to store single branches in the AFOPT structure, which can lead to significant space saving and construction cost saving.

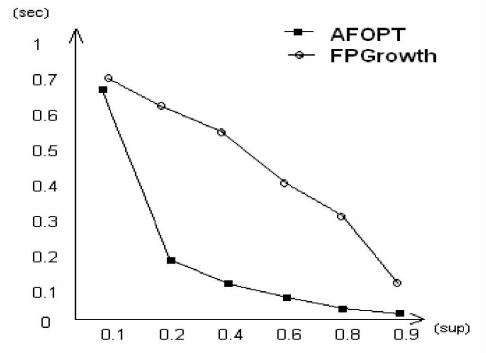


Figure 6: AFOPT vs. FP-Growth

The push right step is needed because otherwise, an item's conditional database would consist of multiple subtrees, the number of which is exponential to the number of items before that item in worst case, while the number of merging operations needed is equal to the number of items before that item in worst case. To save the traversal cost, the merging operation is performed. The items are sorted in descending frequency order in FP-Tree, which improves the possibility of prefix sharing. Hence FP-Tree is more compact than the AFOPT structure.

Applying the algorithm at chapter 3 on an AFOPT structure will be about the same as applying the transformation on an FP-Tree, because the same (maximal) frequent patterns are extracted, the only difference being the order (reversed). On the other hand, as stated earlier, the AFOPT algorithm needs less traversal cost than the FP-Growth algorithm. Thus, the use of the AFOPT algorithm is more recommended (fig.6).

The AFOPT algorithm traverses the trees in top-down depth-first order, and the items in the prefix-trees are sorted in ascending frequency order. The combination of these two methods is more efficient than the combination of the bottom-up traversal strategy and descending frequency order, which is adopted by the FP-Growth algorithm. Both of the above two combinations are much more efficient than the other two combinations - the combination of the top-down traversal strategy and the descending frequency order, and the combination of the bottom-up traversal strategy and the ascending frequency order.

Experiments were conducted on a 1.6 Ghz Athlon XP with 256 MB memory running Microsoft Windows XP Professional using a real database of 50000 entries in 6000 transactions. All codes were done in Java and compiled using the Eclipse Platform.

References

[1] Agrawal R. and Srikant R., "Fast algorithms for mining association rules in large databases". *Proc. of 20th Int'l conf. on VLDB*: 487-499, 1994.

[2] Agrawal R., Agrawal C. and Prasad V.V.V., "Depth first generation of large itemsets for association rules", *IBM Tech. Report*, RC21538, 1999

[3] Han J., Pei J., Yin Y., "Mining Frequent Patterns without Candidate Generation". *Proc. of ACM-SIGMOD*, 2000

[4] Han J. and Kamber M., "Data Mining Concepts and Techniques", *Morgan Kaufmann Publishers, San Francisco, USA, ISBN 1558604898*, 2001.

[5] Han J. and Fu Y, "Discovery of multiple-level association rules from large databases" *Proceeding of the International Very Large Databases Conference*, pg. 420-431, 1995

[6] Runying Mao, "Adaptive–FP: An Efficient and Effective Method for Multi–Level and Multi–Dimensional Frequent Pattern Mining", Simon Fraser University, April 2001

[7] Klemettinen M., Mannila H., Ronkainen P., Toivonen H., and Verkamo A.I. (1994), "Finding interesting rules from large sets of discovered association rules", *In CIKM'94*, pp. 401-408.

[8] Rajkumar N., Karthik M.R. and Sivanandam S.N., "Fast Algorithm for Mining Multilevel Association Rules", *IEEE Web Technology and Data Mining*, pg. 687-692, TENCON, 2003

[9] Srikant R. and Agrawal R., "Mining Generalized Association Rules", *In Research Report RJ 9963*, IBM Almaden Research Center, San Jose, California, USA, June 1995

[10] Liu G., Lu H. and Lou W., „Efficient Mining of Frequent Patterns Using Ascending Frequency Ordered Prefix Tree", *Data Mining and Knowledge Discovery*, 9, 249–274, 2004

[11] Liu G., Lu H., Lou W., Xu Y. and Xu Yu J., "Ascending Frequency Ordered Prefix-tree: Efficient Mining of Frequent Patterns" *In Proc. of KDD Conf.*, 2003.

[12] Győrödi R., Győrödi C., Pater M., Boc O., David Z., "FP-Growth algorithm for multi-level databases", *CSCS-15 15th International Conference on Control Systems and Computer Science*, Bucuresti, 2005

[13] Győrödi R., Győrödi C., Pater M., Boc O., David Z., "AFOPT Algorithm for multi-level databases", *SYNASC 05*, Timisoara, 2005

[14] Pater M., Győrödi R., Győrödi C., Boc O., David Z., "Fast Apriori algorithm for multilevel database", *microCAD 2005, International Scientific Conference*, pg. 347-352, Miskolcs, Hungary, 10-11 March 2005

[15] Pater M., Győrödi R., Győrödi C., Boc O., David Z., "Multi-level Apriori Algorithms", *CSCS-15 15th International Conference on Control Systems and Computer Science*, Bucuresti, 2005

Mirela Pater, Cornelia Győrödi, Robert Győrödi, Alina Bogan-Marta
University of Oradea
Department of Computer Science
Address: 1, Universitatii St, Oradea, Romania
E-mail: {mirelap,cgyorodi,rgyorodi,alinab}@uoradea.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 349-354

Operational Security Metrics for Large Networks

Victor-Valeriu Patriciu, Iustin Priescu, Sebastian Nicolăescu

Abstract: Managing the security of large networks has become a critical issue in the era of Internet economy. As any other process, security can not be managed, if it can not be measured. The need for metrics is important for assessing the current security status, to develop operational best practices and also for guiding future security research. This paper presents some metrics used by large ISP-es to monitor the level of security in their networks.

Keywords: network security, metrics, vulnerabilities, risk, monitoring

1 Introduction

The current strategies for evaluating or validating IT systems and network security are focused on:

• Examining the results of security assessments, including red-teaming exercises, penetration testing, vulnerability scanning, and other means of probing defenses for weaknesses in security, and

• Examining the building blocks, processes, and controls associated with security efforts to infer the prevalence of vulnerabilities. Activities include auditing business processes and procedures for security policy compliance, assessing the quality of security in infrastructure components, and reviewing system development and administration processes for security best practices.

These measurement strategies are not good enough considering higher frequency the new vulnerabilities are identified, and the shorter interval the exploit becomes available to the attackers after the vulnerability is publicly announced. As practice showed that any prevention mechanism may fail, a real-time security monitoring strategy and a set of good metrics would help both to determine the status of IT security performance, and to enhance it by minimizing the windows of exposure to the new vulnerabilities.

IT security metrics can be obtained at different levels within an organization. Detailed metrics, collected at the system and network level, can be aggregated and rolled up to progressively higher levels, depending on the size and complexity of an organization. If measurements are instantaneous snapshots of a particular measurable parameters, then metrics are more complete pictures, typically comprised of several measurements, baselines, and other supporting information that provide context for interpreting the measurements. Good metrics are goal-oriented and should have the following features: *specific*, *measurable*, *comparable*, *attainable*, *repeatable*, and *time dependent*.

2 Standardization - Drivers and Results

Security performance measurement by using standardized metrics gained increasingly interest during the last years with the help of guidelines, code of practices and standards accepted widely over the world, and with the efforts of international organizations and companies. Code of practices like BS7799, ISO17799, NIST SP800-33 are useful as a starting point for security measures in organizations. They focus mainly on providing sets of controls, but the measurement of the quality and applicability of these controls is not handled in detail. In 2004, Security Metrics (SECMET) Consortium was founded to define standardized quantitative security risk metrics for industry, corporate and vendor adoption by top corporate security officers of the sector. Another standardization effort is led by the Metrics Work Group of International Systems Security Engineering Association (ISSEA). This group is tasked to develop metrics for Systems Security Engineering - Capability Maturity Model (SSE-CMM). SSECMM has adopted the NIST 800-55 methodology of developing security and process metrics. The work group has proposed 22 Process Areas (PA) for metrics development grouped in two sections: security base practices and project and organizational base practices.

Meanwhile, governments around the world already released laws and regulations driving and facilitating IT security measurements. Some example of laws and government regulations are: Gramm-Leach-Bliley Act (GLBA), the Health Insurance Portability and Accountability Act (HIPAA), the Federal Information Security Management Act of 2002 (FISMA) - for the US, and The Data Protection Directive 95/46/EC of the European Parliament - for the EU.

The most important methods used to develop security metrics are: the IT performance assessment methodology,

the stakeholder-based model and the capability-based model. *Capability-based model* is a product of SSE-CMM international metric project. It addresses the functional capabilities: protect, detect, and respond. SSE-CMM defines required performance of the best practices to generate specific results. *IT performance assessment methodology* (coordinated by US Department of Defense) has three components: capabilities, attribute level, and specific metrics. The attribute level addresses the requirement that support that mission and the specific metrics component addresses specific measurable activities that support those mission requirements. The *stakeholder-based model* views metrics from an organizational role perspective: stockholders, stockholders responsibility, stockholders interest and actions.

The challenge of defining security metrics lies on the problem that metrics must be quantifiable information (like percentage, average or absolute numbers) for comparison, applying formulas for analysis and tracking the changes. The result from the manual collection or automated resources should be meaningful performance data and must be based on IT security performance goals of the organization. Metrics should also be easily obtainable and feasible to measure. But research methodology plays an important role here, not to have biased data as a result; and to cover all dimensions of IT security from organizational (people), technical and operational points of view.

3 Metrics to Evaluate the Network Security Vulnerabilities

CERT reported in 2005 a number of 5,990 vulnerabilities, which represents an increase with 58% from 2004. To determine the urgency and priority of response to vulnerabilities, organizations need models that would convey vulnerability severity.

One such model is the Common Vulnerability Scoring System (CVSS), and it was designed to provide the end user with an overall composite score representing the severity and risk of a vulnerability. The score is derived from metrics and formulas. The metrics are in three distinct categories that can be quantitatively or qualitatively measured. *Base metrics* contain qualities that are intrinsic to any given vulnerability that do not change over time or in different environments. *Temporal metrics* contain vulnerability characteristics which evolve over the lifetime of vulnerability. *Environmental metrics* contain those vulnerability characteristics which are tied to an implementation in a specific user's environment. The particular constituent metrics used in CVSS were identified as the best compromise between completeness, ease-of-use and accuracy. They represent the cumulative experience of the model's authors as well as extensive testing of real-world vulnerabilities in end-user environments.

There are seven *base metrics* which represent the most fundamental features of vulnerability:

• *Access Vector* (AV) measures whether the vulnerability is exploitable locally or remotely.

• *Access Complexity* (AC) measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target system. The set of values is: high or low.

• *Authentication* (A) measures whether or not an attacker needs to be authenticated to the target system in order to exploit the vulnerability. The set of values is: required or not required.

• *Confidentiality Impact* (CI) measures the impact on confidentiality of a successful exploit of the vulnerability on the target system. The set of values is: none, partial or complete.

• *Integrity Impact* (II) measures the impact on integrity of a successful exploit of the vulnerability on the target system. The set of values is: none, partial or complete.

• *Availability Impact* (AI) measures the impact on availability of a successful exploit of the vulnerability on the target system. The set of values is: none, partial or complete.

• *Impact Bias* (IB) allows a score to convey greater weighting to one of three impact metrics over the other two. The value can be normal (CI, II and AI are all assigned the same weight), confidentiality (CI is assigned greater weight than II or AI), integrity (II is assigned greater weight than CI or AI), or availability (AI is assigned greater weight than CI or II).

The *temporal metrics* which represent the time dependent features of the vulnerability are:

• *Exploit Ability* (E) measures how complex the process is to exploit the vulnerability in the target system. The set of values is: unproven, proof of concept, functional, or high.

• *Remediation Level* (RL) measures the level of an available solution. The set of values is: official fix, temporary fix, workaround, or unavailable.

• *Report Confidence* (RC) measures the degree of confidence in the existence of the vulnerability and the credibility of its report. The set of values is: unconfirmed, uncorroborated, or confirmed.

The *environmental metrics* represent the implementation and environment specific features of the vulnerability.

• *Collateral Damage Potential* (CDP) measures the potential for a loss of physical equipment, property damage

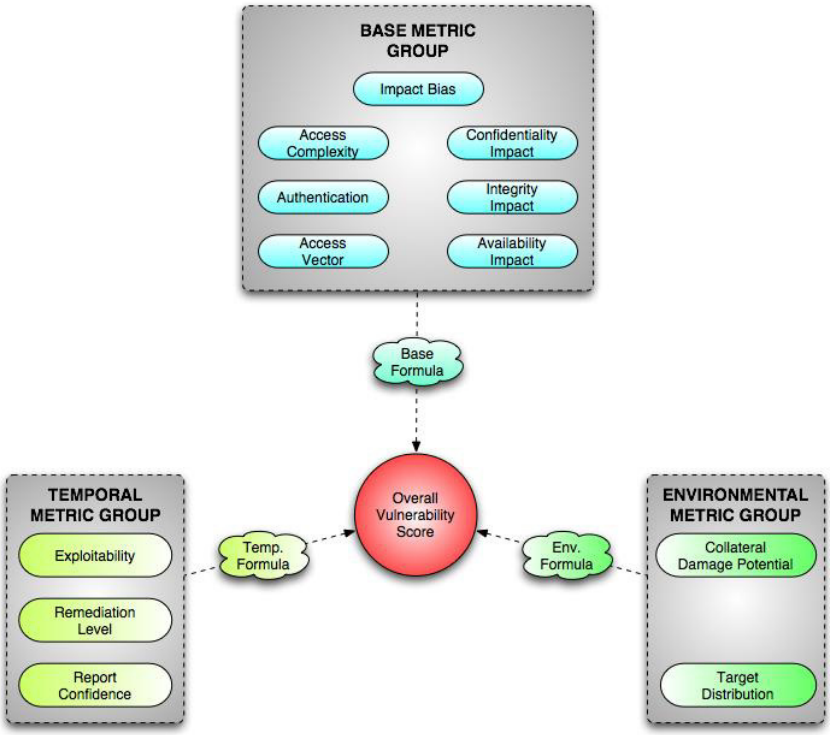


Figure 1: Common Vulnerability Scoring System Framework

or loss of life or limb. The set of values is: none, low, medium, or high.

• *Target Distribution* (TD) measures the relative size of the field of target systems susceptible to the vulnerability. The set of values is: none, low, medium, or high.

Scoring is the process of combining all the metric values according to specific formulas. *Base Score* (BS) is computed by the vendor or originator using the following formula:

```
BS= round (10 * AV * AC * A * ((CI*CIB) + (II*IIB) + (AI*AIB))),
```

Once is set and published, the BS score is not expected to change. It is computed from "the big three" - confidentiality, integrity and availability. This is the "foundation" which is modified by the temporal and environmental metrics. The base score has the largest bearing on the final score and represents vulnerability severity. *Temporal Score* (TS) is also computed by vendors and coordinators for publication based on the following formula:

```
TS= round (BS * E * RL * RC),
```

It allows for the introduction of mitigating factors to reduce the score of the vulnerability and is designed to be re-evaluated at specific intervals as a vulnerability ages. The temporal score represents vulnerability urgency at specific points in time. *Environmental Score* (ES) is optionally computed by end-user organizations and adjusts combined base-temporal score based on the following formula:

```
ES= round ((TS + ((10 - TS)* CDP))* TD),
```

This should be considered the *final score* and represents a snapshot in time, tailored to a specific environment. User organizations should use this to prioritize responses within their own environments. CVSS differs from other scoring systems (e.g. Microsoft Threat Scoring System, Symantec Threat Scoring System, CERT Vulnerability Scoring or SANS Critical Vulnerability Analysis Scale Ratings) by offering an open framework that can be used to rank vulnerabilities in a consistent fashion while at the same time allowing for personalization within each user environment. As CVSS matures, these metrics may expand or adjust making it even more accurate, flexible and representative of modern vulnerabilities and their risks.

4 Metrics to Evaluate the Network Security Controls

In most large organization, measurements of network security are often conducted by separate teams that independently define, collect, and analyze technical metrics. These metrics include the numbers of vulnerabilities found in network scans, known incidents reported, estimated losses from security events, security bug discovery rate in a new software application, intrusion detection system alerts, number of virus infected e-mails intercepted, and others.

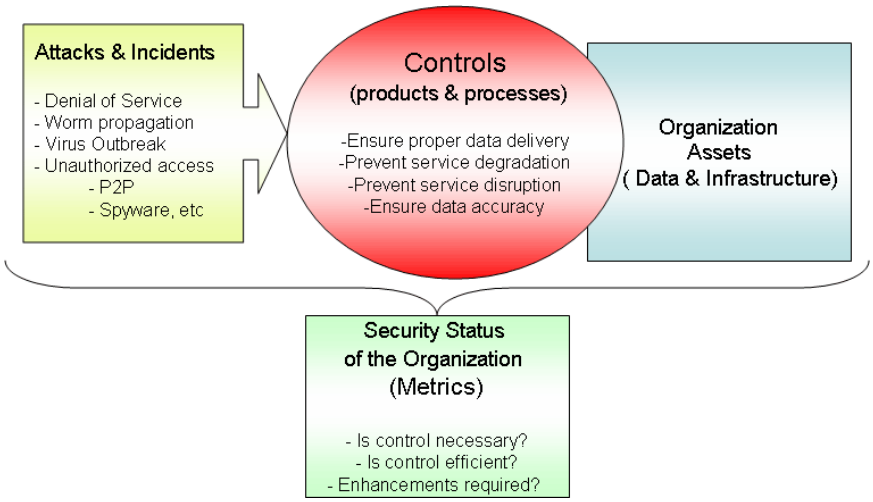


Figure 2: Network Security Based Upon Metrics

The security metrics described in this section focus on network integrity and reliability. The other aspects like network information asset value, loss, and opportunity cost are not subject of this presentation. Depending upon their role in interacting with the network (stakeholder-based model), various users are concerned about different aspects of network security.

Executive officers, being responsible for the overall performance of the enterprise, are concerned with the ability of the network to support operations. Because they have the authority to allocate resources, both personnel and financial, to deal with problems of network security, they would be interested in answers to the questions like: How does the organization's network security compare to that of similar organizations? How does network security this year compare to last year? Does the security spending generate the expected return? What are the costs and consequences of not acting to improve network security?

An example of the network security metrics used at the management level is:

• *Network Service Level* - Percentage of time that network services are available for a given period of time as well as part of a timeseries to give historical context.

• *Business Requirements Met* - Percentage of business needs supported by the network infrastructure and which are being met.

• *Number of Compromises* - Number of incidents during a given period in which network security was compromised.

• *Organizational Impact of Compromises* - For each incident, the number of hours, time of day, and people affected by the degradation or disruption of network service.

• *Costs and Benefits of Improvements* - The direct and indirect costs and benefits of steps that can be taken to improve network security.

• *Peer Performances* - Network service level benchmarks from similar organizations.

Network and IT systems operations groups, responsible for infrastructure, and systems production support, are generally interested in a more granular view of the network security to maintain network services. Whereas executives look for support for resource allocation decisions, network operations people seek help to prevent, detect, and respond to network security intrusions. Thus, questions of concern include: What computers, applications, or services are compromising network security? Where are they? How is the compromise taking place? Is it getting worse? How and where? How serious is the impact of the compromise? What technical measure can be taken to

isolate and remediate the problem machines?

An example of the network security metrics used by network and IT operation groups is:

- *Compliant Devices* - Percentage of network devices that are security policy compliant.
- *Managed Devices* - Counts of systems and devices under active management.
- *Total Devices and Users* - Total numbers of devices and users on the network.
- *Network Latency* - Mean time for packet delivery in the network.
- *Packet Loss* - Percentage of packet losses.
- *Network Utilization* - Bandwidth utilization at key gateways in the network.
- *Network Throughput* - Transfer rate for defined end-to-end network services, such as FTP, POP3, HTTP, etc.
- *Viruses Detected in e-mail Messages* - percentage of emails infected by viruses.
- *Unauthorized Accesses Attempts* - percentage of unauthorized access for various network services (VPN, HTTP, SSH, etc.)
- *Impact of Compromise* - Users affected (service degraded, disrupted, or otherwise compromised), number of devices participating in compromise, decrease in network performance, increase in network utilization, and increases in wait times during a network compromise.

The *network security team* is typically responsible for the organization's security policies and programs. Although they may not have direct operational responsibility, they are interested in how security policies, procedures, and programs are ensuring or failing to ensure network security. Questions of concern include: Were the computers responsible for compromising the network policy compliant? What changes should be made to security policies and procedures? If policies are not working, what behavior changes should policy modifications be aiming to achieve? What technologies could help prevent future compromises? What was the impact of the compromise?

A sample of the network security metrics used by security operation team is available below:

- *Vulnerability Counts* - Numbers of vulnerabilities found on the network, broken out by those on policy-compliant devices vs. those found on devices that are not.
- *Intrusion Attempts* - Number of true/false positive/negative intrusions attempts.
- *Unauthorized Accesses Attempts* - percentage of unauthorized access for various network services (VPN, HTTP, SSH, etc).
- *Detailed Compliance Reports* - Numbers of users and devices compliant with each element of the security policy.
- *Incident Forensics* - The numbers of incidents attributable to policy failures vs. policy compliance failures.
- *Impact of Compromise* - Users affected (service degraded, disrupted, or otherwise compromised); data lost, modified, or destroyed; number of devices participating in compromise; decrease in network performance; increase in network utilization; and increases in wait times during a network compromise.
- *Suspect Port Scans* - Number of suspect scans on organization's network (eg. requests sent on port 80 to routers are suspect).
- *Remediation Time* - Time between compromise discovery and completion of system remediation.

The measurement process can be automated by implementing the network security monitoring solutions. In this way, measurement errors and the subjective interpretations are eliminated, making possible for credible measurement comparisons across either time (time-series) or organizations (benchmarks).

5 Summary and Conclusions

Security metrics are tools designed to facilitate decision making and improve performance and accountability through collection, analysis, and reporting of relevant security performance-related data. Security metrics, at least such metrics trying to define a measure for the security of an entire organization, are a quite new area of research. The paper presented the major standardization players and results in this area.

Given the increased number of vulnerabilities the organizations have to handle, we presented an open source framework (CVSS) that can be used to rank vulnerabilities in a consistent fashion while at the same time allowing for personalization within each user environment.

In the last section we covered the metrics for network security from the perspective of stakeholder based model, and presented the major technical-operational metrics used by major network operators.

References

[1] Andrew Jaquith, *Security Metrics: Replacing Fear, Uncertainty, and Doubt*, Addison Wesley, 2006.

[2] Gerald L. Kovacich, Edward Halibozek, *Security Metrics Management: How to Measure the Costs and Benefits of Security*, Butterworth-Heinemann, 2005.

[3] Marianne Swanson P & others, "Security Metrics Guide for Information Technology Systems," *NIST Special Publication 800-55*, 2003 (http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf).

[4] Ron Ross, & others, "Recommended Security Controls for Federal Information Systems," *NIST Special Publication 800-53*, 2005 (http://csrc.nist.gov/publications/nistpubs/800-53/SP800-53.pdf).

[5] Systems Security Engineering-Capability Maturity Model Group, *SSE-CMM - Model Description Document version 3.0*, International Systems Security Engineering Association, 2003.

[6] Mike Schiffman, Cisco CIAG, "A Complete Guide to the Common Vulnerability Scoring System (CVSS)," *Forum Incident Response and Security Teams*, 2005. (http://www.first.org/)

[7] VV Patriciu, I. Priescu, S. Nicolăescu, "Security Monitoring - An Advanced Tactic for Network Security Management," *The 6th IEEE Communications International Conference*, Bucharest, 2006.

[8] ISO/IEC, *Information Technology - Code of practice for information security management (final draft)*, ISO, 2005.

[9] British Standard Institute, *Information Security Management. Code of Practice for Information Security Management (BS 7799-1)*, British Standard Institute, 1999.

[10] CERT, *CERT/CC Statistics 1988-2005*, CERT, 2005 (http://www.cert.org/stats/).

Victor-Valeriu Patriciu, Iustin Priescu
Military Technical Academy, Department of Informatics & Mathematics
Address: Bd. George Cosbuc, 81-83, sector 5, Bucharest, Romania
E-mail: {vip, iustin}@mta.ro

Sebastian Nicolăescu
MCI - Enterprise Production Services, USA
E-mail: sebastian.nicolaescu@mci.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 355-359

An Interactive Learning Environment for Analyze Linked List Data Structures

Manuela Pănoiu, Caius Pănoiu, Ionel Muscalagiu, Anca Elena Iordan

Abstract: Informational society needs major changes in the educational programs, being necessary to prepare teachers from all fields, in such way to use the information technologies in computer assisted learning. A well-known term for this field is CBL (Computer Based Learning). The computer, as auxiliary element of the teacher, is used since few decades, but the new technologies appeared in computer science lead to the necessity to adapt the learning/teaching methodologies in such way to be based increasingly on information technology. This paper presents a software package, which can be used as educational software. This paper present a graphical user interface implements in Borland Delphi useful for computer based learning. It is made a study of linked list.

Keywords: e-learning, educational software, linked list

1 Introduction

Today, it is use the computer in many areas, including in education, from the elementary school to the universities, by following some purposes. Between these purposes it can be cite the necessity to prepare the young people for a future where the computer science is a dominant area. This purpose can be realized by using new didactical teaching methods based on educational software that is programs useful in teaching-learning process [1][2] [8]. The CBL type systems (Computer Based Learning) are used in universities from the entire world. For instance, at Dundee University, Scotland, UK [2] a CBL system is integrated part in the courses from the first two years in computers field. These courses present the students the object-oriented programming, data structures and algorithms. Has been made study on a group of 8 students from 35 to evaluate the CBL methods. Further this study, were drawn the following conclusions: Importance of the design (an attractive and clear interface), the role of CBL as support for the course, in order for this to be able to replace totally the classic support and the role of CBL as motivator - the students having the possibility to interact with virtual created environment, this method being preferred in comparison with lecturing of a classic material. By using the computer in didactical activity it is increase the learning productivity: the necessary information is faster and accurate obtained, process and sends, by eliminate unnecessary time delay. It can be showing to the student some information which is inaccessible otherwise: dynamic diagrams, moving images, sounds. The computer can quickly process all the available information, by a relevant manner present. The educational software has an important characteristic: it is and can be interactive. Thus, the user can interfere with the program by changing on line some parameters for the studies process. Technologies used to create virtual environments are very diverse [7], [9],[10],[11], and are used with success in diverse areas from the educational field, e.g. in physics, mathematic, architecture, etc. The institutions of superior education, and not only these ones, must comprise as soon as possible the possibilities of new online technologies as quality improvement means for education. This increasing interest for online technologies used in education is accompanied by the increasing interest for different strategies for online monitoring of the courses' quality. The students must be able to use different types of educational environments, as for these to combine flexibly different types of interfaces. Not all types of educational soft packages meet these conditions. Many of the soft packages used in mathematic and sciences excel by the quantity of scientific information, but in return they are poorer from other major viewpoints. The soft is too expensive, fragmented, and does not meet the market's requirements. Teaching programming to novices has proven to be a challenge for both teachers and students. Many students find the programming module difficult and disheartening and this could have an impact on their attitude to software development throughout the course and as a career choice. For staff involved in teaching programming it can also be very disheartening when student apparently fail to understand and be able to use even the basic data structures. These difficulties have prompted researchers to investigate tools and approaches that may ease the difficulty of teaching and learning data structures [4]. The data structures and algorithms are basic elements, fundamentals that must be apprehended by any student in IT, from this reason being a great variety of learning methods, a part of these methods using the computer and being applied in diverse universities worldwide [4], [8]. From desire of improve the instructive educational process, using modern teaching methods, was realized a multimedia courseware where the subject "Linked list" was especially developed.

2 Linked list data structures

One of the most common data structures is the singly linked list. A linked list is a collection of items accessible one after another beginning at the head and ending at the tail. There are implemented by each item having a link to the next item. The head is the first item of a list, and the tail is the last item of the list. An example of a singly linked list is present in fig. 1.

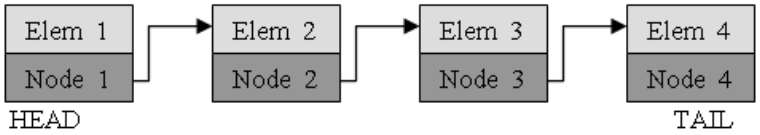


Figure 1: A singly linked list

The doubly linked lists

A variant of a linked list in which each item has a link to the previous item as well as the next. This allows easily accessing list items backward as well as forward and deleting any item in constant time.

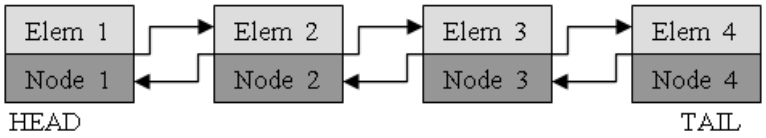


Figure 2: A doubly linked list

The circular linked list

A variant of a linked list in which the nominal tail is linked to the head. The entire list may be accessed starting at any item and following links until one comes to the starting item again.

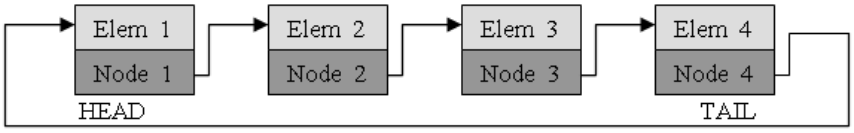


Figure 3: A circular linked list

3 Aplication present

The application is implemented in Borland Delphi 6.0, under Microsoft Windows operating system. The graphical user interface was structured in two parts: a theoretical presentation part and a simulation part. The simulation part contains simulations for all three types of linked list.

From the main application window, present in fig. 4, it can select by a main menu one of these parts.After an option is selected, a new window is opened. A first option was implement for theoretical concepts about dynamic memory allocation and linked lists. The user can read this theoretical part, and it can print this or view a simulation, which are included in following three options from the main menu: singly linked list, doubly linked list and circular linked list.

For each kind of list there are implement an interactive simulation because in interactive training simulations the learner interacts with a system in ways which change or modify the behaviors of the simulation. The singly linked list option was developed in another window, show in fig. 5.

All the common operation with the singly linked list is presented in this window. The student has the possibility, in an interactive mode to select and understand these operations. One of the most important characteristic of a courseware is the possibility to be interactive. By click the "Start creating" button an empty list is created, as is show in figure 5. After that, the user can add the nodes successively. It is also possible to delete a node from the

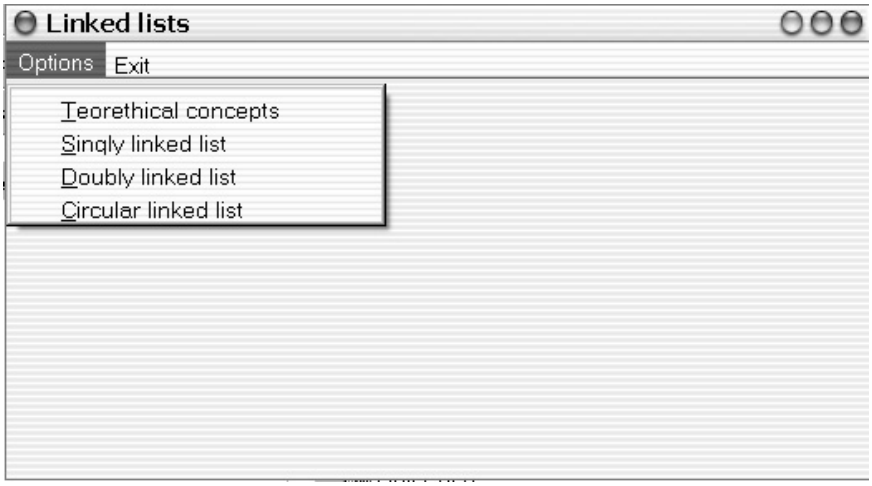


Figure 4: The main application window

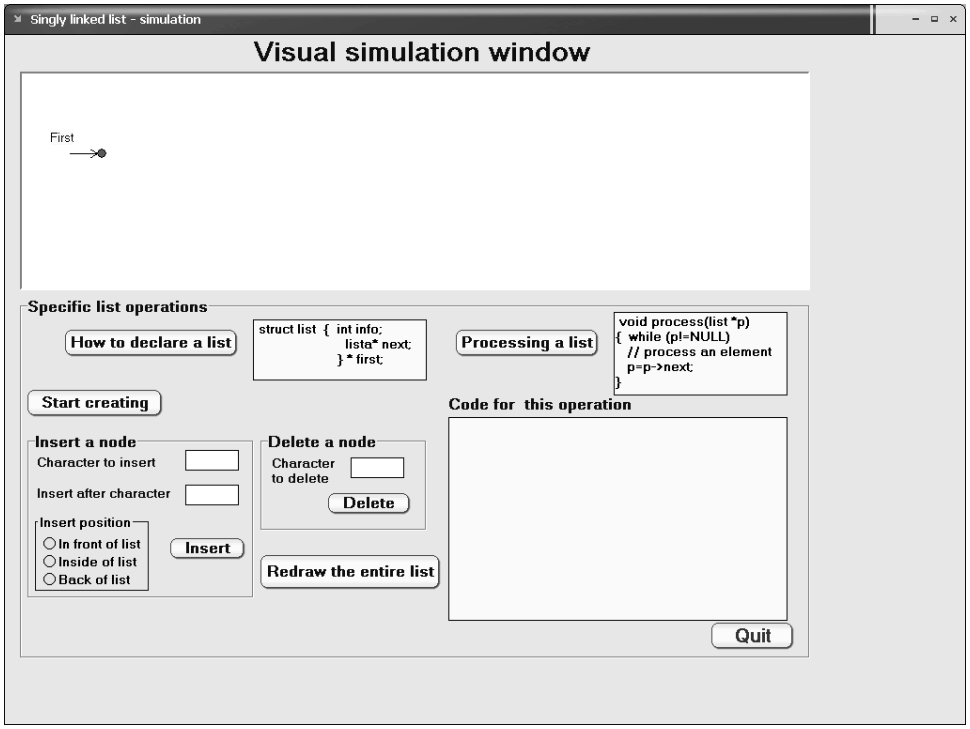


Figure 5: Singly linked list specifically operation

list. The node which is inserting or deleted is displayed with red color for more accuracy. Inserting a node can be made in three positions: in front of the list, inside of the list or back of the list. For all the three cases a simulation was implementing as is show in Fig. 6. For each operation, the code is show, step by step, at one time with the simulation, as is showing in figure 6 a, b, c, d.

In case that it is necessary, the whole list can be deleted, using the "Start creating" button, and the create process can start at the beginning. One of the most common operations is processing a list, which means processing all the nodes successively. This operation is simulating by using the button "Processing the list" and the source code (in C language) are displayed. In the same manner the "Doubly linked list" and the "Circular linked list" option from the main window was implemented.

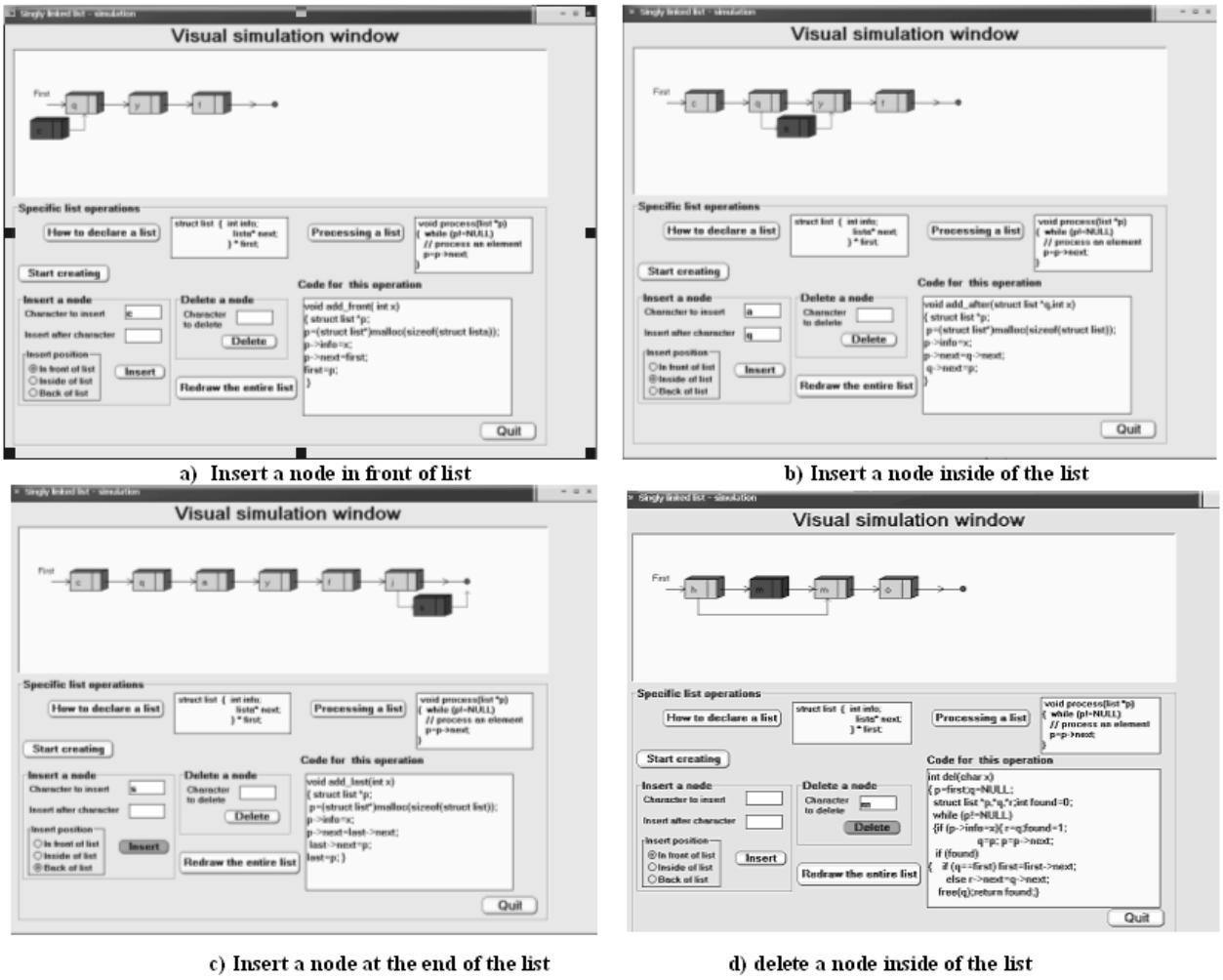


Figure 6: List operation: insert a node: in front of list (a); inside of list (b); at the end of the list (c) and delete a node (d)

4 Summary and Conclusions

On this application, authors take into consideration the condition which must accomplish a courseware, being made necessary steps [1] [2]. So, in elaboration and utilization of this application must take into consideration next criteria: - To accomplish some teaching and learning strategy. In this kind of self-instruction and evaluation program it must find basic notions and representation and scanning notions. Animation and graphical modeling must represent the graphical construction way and also scanning of them; - To exist the possibility to use parameterized variable, in conditions in which users have the possibility to input the variables value; - To present a method in which the user can be informed about how can use graphical module, i.e. an interaction user-computer

exist. The presented application accomplishes these criteria, and for this we consider that is an example of how educational software must be realized.

References

[1] McDougall, A., Squires, D., "Empirical study of a new paradigm for choosing educational software" *Computer Education*, Vol. 25, no. 3, Elsevier Science, 1995.

[2] Glenn, W., Rowe, Gregor, P., "A computer based learning system for teaching computing: implementation and evaluation" *Computer and Education*, Elsevier Science, 1999.

[3] Tennenbaum, Langsam & Augenstein, "Data Structures Using C and C++" *Prentice Hall Publication* , 1996.

[4] Hanciles, B., Shankaraman, V., Munoz, J, "Multiple representations for understanding data structures" *Computer Education* vol 29. no. 1, Elsevier Science, 1997.

[5] Knuth D. E., "Fundamental Algorithms " Third Edition *Massachusetts: Addison-Wesley*, 1997.

[6] Knuth D. E., "Sorting and Searching " Second Edition *Massachusetts: Addison-Wesley*, 1997.

[7] Kurtz, B. , Johnson, D., "Using simulation to teach recursion and binary tree traversals" *Technical Symposium on Computer Science Education, Proceedings of the sixteenth SIGCSE technical symposium on Computer science education* New Orleans, Louisiana, United States,pp. 49 - 54,1985, ISSN:0097-8418.

[8] Scanlon, E., Tosunoglu, C., Jones, A., P. Butcher, S. Ross, J. Greenberg, J.,"Learning with computers: experiences of evaluation" *Computer Education* , Elsevier Science, 1998.

[9] Silveira, R., Viccari, R.,"JADE - Java Agents for Distance Education Framework" *8th Annual International Distance Education Conference* , January 23-26, 2001.

[10] Geissinger, H.,"Educational software: Criteria for Evaluation"
http://www.ascilite.org.au/conferences/perth97/papers/Geissinger/Geissinger.html.

[11] Trindade, J., Fiolhais, C. & Almeida, L. ,"Science Learning in Virtual Environments" *British Journal of Educational Technology* , 33, 4, pp 471-488,2002.

[12] Michailidou, A. & Economides, A. ,"E learn: Towards a Collaborative Educational Virtual Environment" *Journal of Information Technology Education* , vol.2, 2003.

Manuela Pănoiu, Caius Pănoiu, Ionel Muscalagiu, Anca Iordan
Technical University of Timisoara, Engineering Faculty of Hunedoara
Electrotechnical Department
Address: 5, Revolutiei St., 331115, Hunedoara, Romania
E-mail: {m.panoiu,c.panoiu, mionel}@fih.upt.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 360-365

Fuzzy Set Based on Four-Valued Logic

Vasile Pătraşcu

Abstract: This paper presents an extension of fuzzy set based on Łukasiewicz four-valued logic. This new type of fuzzy set is defined by four scalar functions and will be called four-valued fuzzy set $(FVFS)$.

Keywords: Łukasiewicz four-valued logic, four-valued fuzzy set, logarithmic addition, logarithmic scalar multiplication.

1 Introduction

The fuzzy set was defined by Zadeh and is based on binary description [14]. This is done by a vector with two components. The components are two functions that define the membership and the non-membership. Later, Atanassov has extended the fuzzy set using a vector with three function components: the membership, the uncertainty and the non-membership [1]. The meaning of these three functions is similar to that of the values used in the Łukasiewicz three-valued logic: true, uncertainty and false. These fuzzy sets with a ternary description are known as intuitionistic fuzzy sets. This paper defines a new type of fuzzy set based on a vector with four function components: the strong membership, the weak membership, the weak non-membership and the strong non-membership. The meaning of these four components is similar to the one used in the Łukasiewicz four-valued logic [8], [9] and [10]. Next, the paper has the following structure: Section 2 makes a short presentation of the fuzzy set; Section 3 makes a short presentation of the intuitionistic fuzzy set and section 4 presents the four-valued fuzzy set. Finally, section 5 outlines some conclusions.

2 The fuzzy set

Let X denote a universe of discourse. Then a fuzzy set C in X is defined as set of pairs:

$$C = \{\langle x,\ \mu_C(x)\rangle : x \in X\rangle\} \tag{1}$$

where $\mu_C : X \to [0,1]$ is the membership function of C and $\mu_C(x)$ is the grade of membership of x into C (see [14]). Hence, the grade of non-membership of x into C is automatically equal to $\nu_C(x) = 1 - \mu_C(x)$. The membership function μ_C and the non-membership function ν_C verify the equality:

$$\mu_C(x) + \nu_C(x) = 1 \tag{2}$$

If we define the vector

$$\delta_C(x) = (\mu_C(x), \nu_C(x)) \tag{3}$$

and its scalar components verify (2) then the formula (1) has the following equivalent form:

$$C = \{\langle x,\ \delta_C(x)\rangle : x \in X\rangle\} \tag{4}$$

We will denote a family of fuzzy sets in X by $FS(X)$.
We must emphasise that the value $\mu_C(x)$ represents the logical value of the belonging and it can be any real number between 0 and 1. On the other hand, Zadeh has defined a fuzzy set based on binary logic having two logical values: true and false. Thus, each element $x \in X$ has two marks: one refers to its membership and one refers to its non-membership to the fuzzy set C. There is no third possibility. In the classical logic this is known as the " Law of the Excluded Middle ". In real life however, the negation does not always identify with logical negation. This situation is very common in natural language processing, words computing, etc. Therefore, Atanassov [1] suggested an extension of classical fuzzy set, called the intuitionistic fuzzy set.

3 The intuitionistic fuzzy set

An intuitionistic fuzzy set C in X is given by a set of ordered triples

$$C = \{\langle x, \mu_C(x), \nu_C(x)\rangle : x \in X\} \tag{5}$$

where $\mu_C, \nu_C : X \to [0,1]$ are functions such that $\forall x \in X,\ 0 \leq \mu_C(x) + \nu_C(x) \leq 1$. For each x the numbers $\mu_C(x)$ and $\nu_C(x)$ represent the grade of membership and the grade of non-membership of the element $x \in X$ to C, respectively. For each element $x \in X$, we can compute the so called grade of uncertainty of x in C defined as follows: $\pi_C(x) = 1 - \mu_C(x) - \nu_C(x)$. It is immediately seen that $\pi_C(x) \in [0,1], \forall x \in X$. The function π_C expresses a lack of knowledge of whether x belongs to C or not [1]. If $\pi_C(x) = 0$, $\forall x \in X$ then $C \in FS(X)$. The membership function μ_C, the uncertainty function π_C and the non-membership function ν_C verify the equality:

$$\mu_C(x) + \nu_C(x) + \pi_C(x) = 1 \tag{6}$$

If we define the vector

$$\delta_C(x) = (\mu_C(x), \pi_C(x), \nu_C(x)) \tag{7}$$

and its scalar components verify (6) then the formula (5) has the following equivalent form:

$$C = \{\langle x,\ \delta_C(x)\rangle : x \in X\} \tag{8}$$

We will denote a family of intuitionistic fuzzy sets in X by $IFS(X)$.

4 The four-valued fuzzy set

4.1 The four-valued fuzzy set definition

We can do a new extension of the classical fuzzy set using the four-valued logic. There are two variants for the four-valued logic. The first is the Łukasiewicz logic based on two steps for the true and two steps for the false. The second is the Bochvar logic that uses the true T, the uncertainty U, the contradiction C and the false F [2],[3], [4], [5], [6]. In this paper, the construction of the four-valued fuzzy set takes into account the Łukasiewicz logic [8], [9], and [10]. Now, we consider a nonempty set X as a universe of discourse. Let it be a an arbitrary property, x an element of X and $w(x)$ the logical value of the sentence " x has the property a". Considering the binary logic, $w(x)$ can have two values: $T = 1$ when the sentence is true and $F = 0$ when the sentence is false. Thus, we obtain two crisp sets:

$$W_\mu = \{x \in X \mid\ w(x) = 1\}$$

and its complement

$$W_\nu = \{x \in X \mid\ w(x) = 0\}$$

In other words, $\mu = w$ represents the crisp membership function of the set W_μ and $\nu = 1 - w$ represents the crisp membership function of the set W_ν. Considering the Łukasiewicz four-valued logic, $w(x)$ can have four values: the true T_0, the contingent true T_1, the contingent false F_1 and the false F_0. Usually $T_0 = 1,\ T_1 = \frac{2}{3},\ F_1 = \frac{1}{3},\ F_0 = 0$. Thus we obtain the following four crisp sets:

$$W_\mu = \{x \in X \mid\ w(x) = T_0\}$$
$$W_\lambda = \{x \in X \mid\ w(x) = T_1\}$$
$$W_\omega = \{x \in X \mid\ w(x) = F_1\}$$
$$W_\nu = \{x \in X \mid\ w(x) = F_0\}$$

Let $\mu, \lambda, \omega, \nu : X \to \{0,1\}$ the membership function of these four crisp sets. These functions construct a crisp partition of X and verify the following two equalities:

$$\mu(x) + \lambda(x) + \omega(x) + \nu(x) = 1 \tag{9}$$

$$w(x) = 1 \cdot \mu(x) + \frac{2}{3} \cdot \lambda(x) + \frac{1}{3} \cdot \omega(x) + 0 \cdot \nu(x) \tag{10}$$

Thus, in the framework of the Łukasiewicz four-valued logic, the set W can be defined knowing four crisp membership functions. We can generalize considering four fuzzy membership functions

$$\mu, \lambda, \omega, \nu : X \to [0,1]$$

that verify (9). We denote this by:

$$W = \{\langle x, \mu(x), \lambda(x), \omega(x), \nu(x)\rangle : x \in X\} \tag{11}$$

On this way, we have obtained the four-valued fuzzy set definition. If we denote

$$\delta(x) = (\mu(x), \lambda(x), \omega(x), \nu(x)) \tag{12}$$

then (11) becomes:

$$W = \{\langle x, \delta(x)\rangle : x \in X\} \tag{13}$$

The scalar components of the vector $\delta(x)$ make a nuanced definition for the belongingness of element x to the set W. These four functions could be called as follows: μ the strong membership function, λ the weak membership function, ω the weak non-membership function and ν the strong non-membership function. We will denote a family of four-valued fuzzy sets in X by $FVFS(X)$.

4.2 The basic operations

Let there be two sets: A,$B \in FVFS(X)$. We define the complement, the union, the intersection, the inclusion and the equality.

The complement:

The complement of the set $A \in FVFS$ will be defined by

$$\overline{A} = \left\{\left\langle x, \overline{\delta}_A(x)\right\rangle : x \in X\right\}$$

where $\overline{\delta}_A(x)$ is the negation of $\delta_A(x)$, namely:

$$\overline{\delta}_A(x) = (\nu_A(x), \omega_A(x), \lambda_A(x), \mu_A(x))$$

The union:

$$\begin{cases} \mu_{A\cup B} = \mu_A \vee \mu_B \\ \lambda_{A\cup B} = (\mu_A + \lambda_A) \vee (\mu_B + \lambda_B) - \mu_A \vee \mu_B \\ \omega_{A\cup B} = (\nu_A + \omega_A) \wedge (\nu_B + \omega_B) - \nu_A \wedge \nu_B \\ \nu_{A\cup B} = \nu_A \wedge \nu_B \end{cases} \tag{14}$$

where $\vee$ and $\wedge$ refer to any couple (*s-norm*, *t-norm*) [5].

From (9) we have:

$$(\mu_A + \lambda_A) \vee (\mu_B + \lambda_B) = (1 - \nu_A - \omega_A) \vee (1 - \nu_B - \omega_B)$$

$$(\mu_A + \lambda_A) \vee (\mu_B + \lambda_B) = 1 - (\nu_A + \omega_A) \wedge (\nu_B + \omega_B)$$

$$\mu_{A\cup B} + \lambda_{A\cup B} = 1 - (\nu_{A\cup B} + \omega_{A\cup B})$$

The condition (9) is verified by the vector $\delta_{A\cup B}$.

The intersection:

$$\begin{cases} \mu_{A\cap B} = \mu_A \wedge \mu_B \\ \lambda_{A\cap B} = (\mu_A + \lambda_A) \wedge (\mu_B + \lambda_B) - \mu_A \wedge \mu_B \\ \omega_{A\cap B} = (\nu_A + \omega_A) \vee (\nu_B + \omega_B) - \nu_A \vee \nu_B \\ \nu_{A\cap B} = \nu_A \vee \nu_B \end{cases} \tag{15}$$

From (9) we have:

$$(\mu_A + \lambda_A) \wedge (\mu_B + \lambda_B) = (1 - \nu_A - \omega_A) \wedge (1 - \nu_B - \omega_B)$$

$$(\mu_A + \lambda_A) \wedge (\mu_B + \lambda_B) = 1 - (\nu_A + \omega_A) \vee (\nu_B + \omega_B)$$

$$\mu_{A\cap B} + \lambda_{A\cap B} = 1 - (\nu_{A\cap B} + \omega_{A\cap B})$$

Thus, the condition (9) is verified by the vector $\delta_{A\cap B}$.
The union and the intersection verify the two De Morgan's formulae:

$$\overline{A\cup B}=\overline{A}\cap\overline{B}$$

$$\overline{A\cap B}=\overline{A}\cup\overline{B}$$

The inclusion:

$$A\subset B\Leftrightarrow\begin{cases}\mu_A\leq\mu_B\\ \mu_A+\lambda_A\leq\mu_B+\lambda_B\\ \nu_A+\omega_A\geq\nu_B+\omega_B\\ \nu_A\geq\nu_B\end{cases}\tag{16}$$

From (16) it results:

$$B\subset A\Leftrightarrow\begin{cases}\mu_A\geq\mu_B\\ \mu_A+\lambda_A\geq\mu_B+\lambda_B\\ \nu_A+\omega_A\leq\nu_B+\omega_B\\ \nu_A\leq\nu_B\end{cases}\tag{17}$$

If 0_X and 1_X are defined by $\delta_0=(0,0,0,1)$ and $\delta_1=(1,0,0,0)$ then for $\forall A\in FVFS$,

$$0_X\subset A\subset 1_X$$

The equality:
From (16) and (17) it results the equality definition:

$$A=B\Leftrightarrow\begin{cases}\mu_A=\mu_B\\ \lambda_A=\lambda_B\\ \omega_A=\omega_B\\ \nu_A=\nu_B\end{cases}\tag{18}$$

The second and the third condition from (16), (17), (18) are equivalent because of the property (9). For the sake of symmetry, we kept the both second and third condition.

4.3 Distances between four-valued fuzzy sets

For simplicity, the universe of discourse X is assumed to be finite, i.e. $X=\{x_1,...,x_n\}$. We can calculate dissimilarities $d(x,y)$ between two elements $x,y\in X$ having as reference the set $A\in FVFS$ and distances $D(A,B)$ between two sets A, $B\in FVFS$, using two variants [7], [11]:
firstly, using the **Euclidian** distance:

$$d_E^2(x,y)=(\mu_A(x)-\mu_A(y))^2+(\lambda_A(x)-\lambda_A(y))^2+(\omega_A(x)-\omega_A(y))^2+(\nu_A(x)-\nu_A(y))^2\tag{19}$$

$$D_E^2(A,B)=\frac{\sum\limits_{x\in X}(\mu_A(x)-\mu_B(x))^2+(\lambda_A(x)-\lambda_B(x))^2+(\omega_A(x)-\omega_B(x))^2+(\nu_A(x)-\nu_B(x))^2}{2n}\tag{20}$$

secondly, using the **Hamming** distance:

$$d_H(x,y)=|\mu_A(x)-\mu_A(y)|+|\lambda_A(x)-\lambda_A(y)|+|\omega_A(x)-\omega_A(y)|+|\nu_A(x)-\nu_A(y)|\tag{21}$$

$$D_H(A,B)=\frac{\sum\limits_{x\in X}|\mu_A(x)-\mu_B(x)|+|\lambda_A(x)-\lambda_B(x)|+|\omega_A(x)-\omega_B(x)|+|\nu_A(x)-\nu_B(x)|}{2n}\tag{22}$$

For (20) and (22), there holds, respectively

$$0\leq D_E^2(A,B)\leq 1$$

and

$$0\leq D_H(A,B)\leq 1$$

4.4 The logarithmic operations

We will define two operations: scalar multiplication and addition. These two operations have a logarithmic behavior [12], [13].
The logarithmic scalar multiplication between a scalar $\alpha \in (0,\infty)$ and a set $A \in FVFS$:

$$\delta_{\alpha\otimes A} = \left(\frac{(\mu_A)^\alpha}{\varphi(\alpha,\delta_A)}, \frac{(\lambda_A)^\alpha}{\varphi(\alpha,\delta_A)}, \frac{(\omega_A)^\alpha}{\varphi(\alpha,\delta_A)}, \frac{(\nu_A)^\alpha}{\varphi(\alpha,\delta_A)}\right) \tag{23}$$

where

$$\varphi(\alpha,\delta_A) = (\mu_A)^\alpha + (\lambda_A)^\alpha + (\omega_A)^\alpha + (\nu_A)^\alpha \tag{24}$$

The scalar multiplication is associative, namely: $\forall \alpha,\beta \in (0,\infty)$,

$$\alpha \otimes (\beta \otimes A) = \alpha\beta \otimes A$$

The logarithmic addition between a set $A \in FVFS$ and a constant $k = (a,b,c,d)$ having $a,b,c,d \neq 0$

$$\delta_{A\oplus k} = \left(\frac{a\cdot\mu_A}{\psi(\delta_A,k)}, \frac{b\cdot\lambda_A}{\psi(\delta_A,k)}, \frac{c\cdot\omega_A}{\psi(\delta_A,k)}, \frac{d\cdot\nu_A}{\psi(\delta_A,k)}\right) \tag{25}$$

where

$$\psi(\delta_A,k) = a\cdot\mu_A + b\cdot\lambda_A + c\cdot\omega_A + d\cdot\nu_A \tag{26}$$

The logarithmic addition between two sets A, $B \in FVFS$ if $\psi(\delta_A,\delta_B) > 0$

$$\delta_{A\oplus B} = \left(\frac{\mu_A\cdot\mu_B}{\psi(\delta_A,\delta_B)}, \frac{\lambda_A\cdot\lambda_B}{\psi(\delta_A,\delta_B)}, \frac{\omega_A\cdot\omega_B}{\psi(\delta_A,\delta_B)}, \frac{\nu_A\cdot\nu_B}{\psi(\delta_A,\delta_B)}\right) \tag{27}$$

The addition is associative and commutative. The neutral element for the addition is the set $\Theta \in FVFS$ defined by:

$$\delta_\Theta = \left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$$

The set Θ verifies the following equalities:

$$A \oplus \Theta = \Theta \oplus A = A$$

4.5 The cardinality of four-valued fuzzy set

We define the cardinalities of the following fuzzy sets: A_μ, A_λ, A_ω and A_ν.

$$card(A_\mu) = \sum_{x\in X} \mu_A(x)$$

$$card(A_\lambda) = \sum_{x\in X} \lambda_A(x)$$

$$card(A_\nu) = \sum_{x\in X} \nu_A(x)$$

$$card(A_\omega) = \sum_{x\in X} \omega_A(x)$$

The cardinalities of the sets A, $\overline{A} \in FVFS$ are defined by:

$$card(A) = card(A_\mu) + card(A_\lambda)$$

$$card(\overline{A}) = card(A_\nu) + card(A_\omega)$$

5 Conclusions

This paper defines the four-valued fuzzy set using descriptors based on a function-vector with four components. The vector structure is similar to Łukasiewicz's four-valued logic constitution. After that, we defined basic set operations on four-valued fuzzy sets, i.e. union, intersection, complement, logarithmic addition and scalar multiplication, distance and cardinality. Moreover, we defined the inclusion operation of $FVFS$ sets, operation which is a natural extension of Zadeh's definition. A further potential research subject could be the application of these new fuzzy sets within the image processing domain.

References

[1] K. Atanassov, Intuitionistic fuzzy sets, *Fuzzy Sets and Systems* 20, pp. 87-96, 1986.

[2] N. Belnap. How a computer should think, in G. Ryle, editor, *Contemporary Aspects of Philosophy*, pp. 30-56. Oriel Press, 1977.

[3] N. Belnap, A Useful Four-valued Logic, in J. M. Dunn and G. Epstein (eds.), *Modern Uses of Multiple-valued Logics*, Reidel Dordrecht-Boston, pp. 8-37, 1977.

[4] D. A. Bochvar, Ob odnom trechznacnom iscislenii i ego primenenii k analizu paradoksov klassiceskogo rassirennogo funkcionalnogo iscislenija, *Matematiceskij Sbornik*, Vol. 4 (46), pp. 287-308, 1938.

[5] C. Cornelis, G. Deschrijver, E. E. Kerre, Square and Triangle: Reflections on Two Proeminent Mathematical Structures for Representation of Imprecision, *Notes on Intuitionistic Fuzzy Sets* 9(3), p. 11-21, 2003.

[6] M. Delgado, D. Sanchez, M. A. Vila, Fuzzy Three-valued and Four-valued Representation of Imprecise Properties, *Proceedings of the IPMU04 Conference*, Perugia, Italy, July 4-9, 2004.

[7] P. Grzegorzewski, E. Mrowka, Subsethood Measure for Intuitionistic Fuzzy Sets,*Proceedings of the IEEE International Conference on Fuzzy Systems*, FUZZ-IEEE 2004, Budapest, Hungary,25–29 July, 2004.

[8] Gr. C. Moisil, *Old and New essays on non-classical logic*s (Romanian), Ed. Stiintifica, Bucharest, 1965.

[9] Gr. C. Moisil, *Essai sur les logiques non-chrysippienne*s, , Ed. Academiei, Bucharest, 1972.

[10] Gr. C. Moisil, *Lectures on the logic of fuzzy reasonin*g, Ed. Stiintifica, Bucharest, 1975.

[11] J. Kacprzyk, E. Szmidt, Similarity of Intuitionistic Fussy Sets and the Jaccard Coefficient, *Proceedings of the IPMU04 Conferenc*e, Perugia, Italy, July 4-9, 2004.

[12] V. Pătraşcu, Gray level image processing using algebraic structures, *Proceedings of the 11th Conference on applied and industrial mathematics – CAIM'03*, Oradea, Romania, pp.167-172, 2003.

[13] V. Pătraşcu, Color Image Enhancement Using the lrgb Coordinates in the Context of Support Fuzzification, Romanian Academy Journal: *Fuzzy Systems & A.I.-Reports and Letters*, Vol 10, Numbers 1-2, pp. 29-42, 2004.

[14] L. Zadeh, Fuzzy sets, *Information and control*, 8, pp. 338-353, 1965.

Vasile Pătraşcu
Tarom Company
Department of Informatics Technology
Address: Calea Bucurestilor, 224F, Otopeni, Ilfov, Romania
E-mail: vpatrascu@tarom.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 366-370

DIETMIX - A Decision Support System for Diet/Feed Mix Problem

Maria Pârv, Vasile Lupşe, Simona Dziţac

Abstract: This paper presents a decision support system, DietMix, running under Microsoft Windows, and implementing a linear programming model for solving general diet/feed mix problems. The program is part of the LINMOD package.

1 Introduction

There is no single universally accepted definition of a decision support system (DSS). For example, Turban [1988, 1992, and 2001], gives, among others, the following two: (1) A DSS is a form of computer software, designed and operated to model or otherwise represent the structure of a decision problem, and thus allow the user(s) to identify and select a preferred strategy or other course of action from two or more alternatives against a pre-determined set of criteria, and (2) A DSS is a set of systematically organised procedures whereby a decision maker can be assisted in capturing and analysing information relevant to a decision task. Such a process can be used repeatedly if required, and can assist any decision maker to anticipate the probable outcome of any given choice or strategy.

To overcome the lack of such an universally accepted definition, Turban proposes four main characteristics for a DSS: (1) DSS incorporate both data and models; (2) DSS are intended to assist managers in their decision making for tasks or decision problems that may have varying levels of structure; (3) DSS are designed to be used to support and not to replace managerial judgement and (4) DSS are designed with the objective of improving the effectiveness of decisions, they will also, generally also improve the efficiency of decision making through speed and the ease with which decision processes may be repeated at will.

Some decision problems are optimization problems, which can be solved using well-known mathematical models. This paper presents DietMix program, part of the LINMOD computer package, which contains several programs dedicated to solve decision problems that can be modeled using linear programming techniques.

2 Mathematical background

2.1 The linear programming model

The general mathematical programming problem is stated as follows:

maximize (minimize) the function:
$f(\mathbf{x}) = f(x_1, x_2, ..., x_n)$
where $\mathbf{x} = (x_1, x_2, ..., x_n)$ **is subject to**:
$g_i(\mathbf{x}) = g_i(x_1, x_2, ..., x_n) \leq 0, i = 1, 2, ..., m.$

The variables $\mathbf{x} = (x_1, x_2, ..., x_n)$ are known as *decision variables*, the function f is called *objective function*, and $g_i (i = 1, 2, ..., m)$ are *constraints*. When the objective function and the constraints are linear, the problem becomes the *linear programming problem*:

maximize (minimize) the function:
$f(\mathbf{x}) = \mathbf{c}\ \mathbf{x} = \sum_{k=1}^{n} c_k x_k$
subject to:
$g_i(\mathbf{x}) = \sum_{k=1}^{n} a_{ik} x_k - b_i \leq 0, i = 1, 2, ..., m.$

The well-known method for solving linear programming problems is the Simplex algorithm [Dantzig, 1963].

2.2 Diet/Feed mix problem

One of the first optimization problems to be studied back in the fourth decade of the last century was the diet mix problem. At that time, U.S. Army wanted to find a diet fulfilling all nutritional requirements of the field GI's and of the minimum cost. George Stigler, one of the first researchers who studied this problem, made in 1939 a

Nutrient	**Nutritional principles**			Cost
1	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	C_1
2	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	C_2
3	$P_{3,1}$	$P_{3,2}$	$P_{3,3}$	C_3
4	$P_{4,1}$	$P_{4,2}$	$P_{4,3}$	C_4
Minimal daily requirement	Z_1	Z_2	Z_3	

Table 1: Diet Mix Data

good "optimal" solution using a heuristic method. Stiegler's guess for the yearly cost was $39.93. Eight years later, in 1947, the first large scale computation in optimization was made by the mathematician Jack Laderman of the Mathematical Tables Project of the National Bureau of Standards. Laderman found an optimal solution of $39.69, using the newly-born simplex method to solve Stiegler's model consisting of 9 equations in 77 unknowns. The computations were performed by 9 clerks during 120 man hours and using hand-operated desk calculators (see, for details, [9]). Diet mix problem involves composing a diet (quantities of available nutrients) for a human or animal such that (1) the cost of the diet to be minimum and (2) the diet maintains the nutritional characteristics above certain limits.

I. *Problem statement*. A cattle farm uses four different nutrients to be incorporated in cattle diet. Each unit of nutrient contains a well-known quantity of the three nutritional principles. The total volume of the diet needs to weight at least K kg. All the input data are shown in the table 1 below, where:

- $P_{i,j}$ represents the quantity of nutritional principle j contained in a unit of nutrient i.
- Z_j denote the minimal daily requirement of nutritional principle j
- C_i is the unit cost of nutrient i.

II. *Notations*. Let

- x_i be the number of units of nutrient i contained in the diet (kg);
- p_j be amount of the nutritional principle j contained in the diet (g);
- g be the weight of the diet (kg);
- c the cost of the diet.

Using the above notations,

- the amounts p_j are computed as follows:
  $p_1 = P_{1,1}x_1 + P_{2,1}x_2 + P_{3,1}x_3 + P_{4,1}x_4$
  $p_2 = P_{1,2}x_1 + P_{2,2}x_2 + P_{3,2}x_3 + P_{4,2}x_4$
  $p_3 = P_{1,3}x_1 + P_{2,3}x_2 + P_{3,3}x_3 + P_{4,3}x_4$,
- the diet weight is: $g = x_1 + x_2 + x_3 + x_4$, and
- the total cost is computed as $c = C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4$.

III. *Constraints* refer to:

- *diet quality*: the amounts of nutritional principles need to be greater than or equal to the minimal daily requirements:
  $p_1 \geq Z_1$, or $P_{1,1}x_1 + P_{2,1}x_2 + P_{3,1}x_3 + P_{4,1}x_4 \geq Z_1$
  $p_2 \geq Z_2$, or $P_{1,2}x_1 + P_{2,2}x_2 + P_{3,2}x_3 + P_{4,2}x_4 \geq Z_2$
  $p_3 \geq Z_3$, or $P_{1,3}x_1 + P_{2,3}x_2 + P_{3,3}x_3 + P_{4,3}x_4 \geq Z_3$.
- *diet weight*: needs to be greater than or equal to K
  $g \geq K$, or $x_1 + x_2 + x_3 + x_4 \geq K$.

- *nonnegativity restrictions* for problem variables:

$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.$

IV. *Objective function*:

$c \rightarrow min$, or
$C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4 \rightarrow min.$

V. *Mathematical programming formulation*.

$C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4 \rightarrow min$
subject to
$P_{1,1}x_1 + P_{2,1}x_2 + P_{3,1}x_3 + P_{4,1}x_4 \geq Z_1$
$P_{1,2}x_1 + P_{2,2}x_2 + P_{3,2}x_3 + P_{4,2}x_4 \geq Z_2$
$P_{1,3}x_1 + P_{2,3}x_2 + P_{3,3}x_3 + P_{4,3}x_4 \geq Z_3$
$x_1 + x_2 + x_3 + x_4 \geq K$
$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 geq 0.$
where all $C_i, Z_j, P_{i,j} (i = 1,2,3,4; j = 1,2,3)$ and K are known quantities.

3 The program

The current version of LINMOD package contains three computer applications: **ProductionPlanning**, **DietMix**, and **Transport**. All are implemented in Microsoft Visual Basic.

3.1 Common user interface features

The graphical user interface of the programs in the LINMOD package is using Romanian language. All programs have some common features with respect to their GUI, as follows:

- input data from keyboard can be stored in a file, and can be reloaded later;
- the results and help are displayed in separate windows.

The above features are implemented by the following elements in the main window:

- separate text boxes for the number of rows and columns;
- a table-like presentation of input data;
- six command buttons:
  - **Incarca** (**Load**) and **Salveaza** (**Save**) - loading and saving the current data in a file;
  - **Prelucreaza** (**Process**) - performing the computations and displaying the results;
  - **Afiseaza** (**Display**) - activating the **Results Window** (see below);
  - **Ajutor** (**Help**) *and*
  - **Inchide** (**Close**).

The **Results Window** contains a text box which fills all its client area and a menu **Fisiere** (**File**). The text box shows all the results produced by the program; the user can scroll in both horizontal and vertical directions.
Its **Fisiere** menu (see Figure 1) has the following menu options:

- **Incarca** (**Load**) - load the content from a text file;
- **Salveaza** (**Save**) - save the content to a text file;
- **Listeaza** (**Print**) - print the content;
- **Inchide** (**Close**).

The programs automatically generate the corresponding mathematical programming (LP) model and then solve it, producing the results in natural language form, using notations specific to the problem being solved. By pressing the **Ajutor** button, the user receives a short description of the problem, in the **Help Window**. The main window provides a range of display options, grouped under the title **Parametri de afisare** (**Display Parameters**).

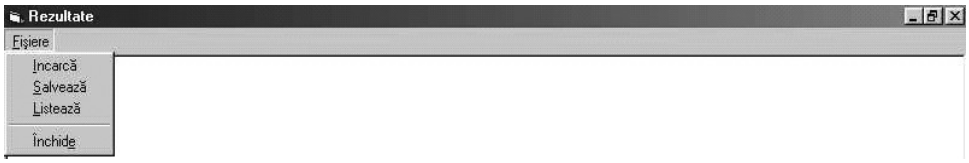


Figure 1: The Results Window with its menu visible

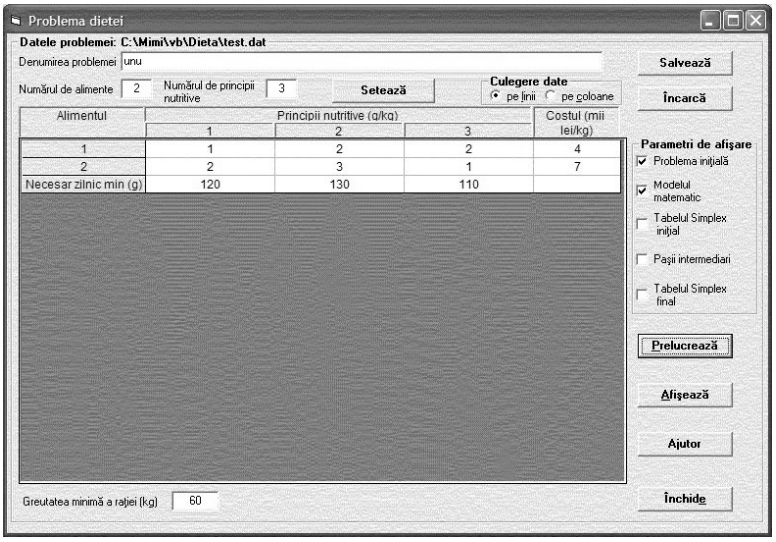


Figure 2: The Main Window of DietMix Program

3.2 The DietMix program

This program builds the corresponding mathematical model of the diet mix problem described in 2.2 and then uses the SIMPLEX algorithm (a variant described in [Marusciac, 1975]) to solve it. Its main window, shown in Figure 2, considers as parameters the number of nutrients and nutritional principles; after this information is supplied by the user, the button **Seteaza** (**Set**) configures the data table corresponding to Table 1.

There are many computer programs or packages implementing linear programming algorithms (see, for example, [Andrei, 2002]). Our intention was not to simply produce a new program offering a better time response or algorithm complexity than the others. Instead, we focused on the simplicity of the user interface, and on its user friendliness. The user sees the same data table as in the problem statement, enters the data in this table, and obtains the results by a simple click, in the Results Window (see Figure 3). Moreover, he/she can obtain several solutions by changing one or more data in the data entry area. Working this way, the mathematical modeling process and the algprithm are hidden; the user operates only with the data and results, using the problem's vocabulary.

4 Conclusions and future work

Our approach can be described shortly as "one problem - one program". Each program in the LINMOD package solves a different LP problem, using the same SIMPLEX algorithm. Correspondingly, the user interface elements, shown in Main Window, Results Window, and Help Window, are belonging to the problem domain/vocabulary.

The programs in the LINMOD package, described extensively in [Pârv, 2004] are intended to ease the way a user uses modelling tools to solve common problems. Pidd (1999) discusses six principles of modelling: (1) simplicity vs complexity, (2) gradual development of models, (3) division of larger models into smaller components, (4) use of analogies, (5) proper use of data, and (6) ordered process. Our opinion is that the programs presented comply with these principles, and help the user to better understand the modelling process.

Our future plans include the development of new programs to be included in the LINMOD package, for solving other decision problems like financial planning, master production schedule, materials requirement planning, factory planning and so on.

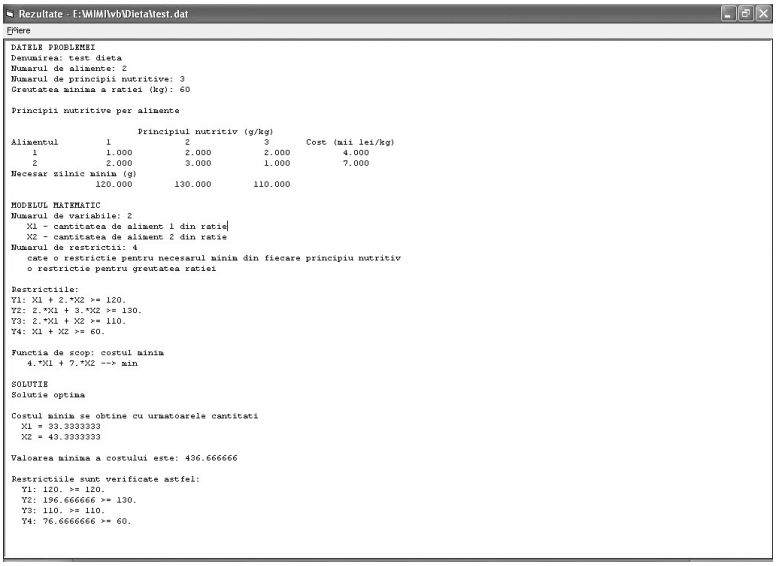


Figure 3: The The Results Window

References

[1] Andrei, Neculai, *Software for Mathematical Programming*, Technical Press, Bucharest, Romania, 2002.

[2] Dantzig, G.B., *Linear Programming and Extension*, Princeton University Press, Princeton, New Jersey, 1963.

[3] Maruşciac, I., *Mathematical programming*, Lito Universitatea "Babes-Bolyai" Cluj-Napoca, Romania (Romanian), 1975.

[4] Pârv, M., *Information systems for the management of the agricultural production and research*, PhD Thesis, University of Agricultural Sciences and Veterinary Medicine Cluj-Napoca, Romania, 2004.

[5] Pidd, M., *Just Modeling Through: A Rough Guide to Modeling*, Interfaces 29, No. 2, 118-132, 1999.

[6] Turban, E., *Decision Support and Expert Systems*, Macmillan, New York, 1988.

[7] Turban, E., *Expert Systems and Applied Artificial Intelligence*, Macmillan, New York, 1992.

[8] Turban, E., Arenson, J., *Decision Support Systems and Intelligent Systems*, Prentice-Hall, 6th ed., 2001.

[9] The Diet Problem: A Brief History of the Diet Problem, www-fp.mcs.anl.gov/otc/Guide/CaseStudies/diet/-history.html

Maria Pârv
University of Agricultural Sciences and Veterinary Medicine
Cluj-Napoca, Romania

Vasile Lupşe
North University
Baia Mare, Romania

Simona Dziţac
University of Oradea
Faculty of Energy Engineering
Address: 1, Universităţii St., 410087, Oradea, Romania
E-mail: sdzitac@rdslink.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 371-375

Applying Task Farming Model over Grids

Dana Petcu, Cosmin Bonchiş, Maria Radu

Abstract: The effectiveness of the task farming model in the case of current Grid environments is studied. A pattern is identified when building Globus-based application. Following this pattern a demo application is developed for real-time application in meteorology.

1 Introduction

The Grid concept has emerged initially to capture the vision of a networked computing system that provides broad access to massive computational resources. This vision has generated interest to many researchers and application developers and new classes of applications arise as being candidate for gridification. A candidate class is that of embarrassingly parallel programs. A farming application fall into this class. A farming application (a pool of tasks or bag of tasks) is one composed of a large number of independent requests that may be served simultaneously [2].

The task farming model (also known as master-slave or master-worker model) has been widely used for developing parallel applications. On a single host or cluster the implementation of the task farming model is quite straightforward. In a Grid environment there are problems with task startup and with communication between the manager (farmer) and the tasks (workers), authentication and authorization to start remote jobs, queues on remote resources, firewalls between resources and so on.

Despite these technical obstacles, task farming applications currently seem to dominate Grid computing. Examples are certain classes of problems require large numbers of nearly identical runs to produce meaningful results, like multiple task farming of gene-sequencing algorithms, Monte-Carlo simulations, data mining, parameter space searches, or analysis of huge volumes of data from large-scale particle physics experiments. Task farming means in these cases launching many simulations at the same time.

Current Grid technologies, like Globus Toolkit (current de-facto standard for Grid computing [3]) for or Nimrod/G (Grid resource broker for managing and steering task farming applications [1]), simplify the development process of an application based on the task farming model.

In what follows we identify the pattern used by a Grid application implementing the task farming model when current Globus technology is used. As a case study a concrete application is described.

2 Task Farming on Globus-based Grids

In the task farming model there are two distinct types of processes: farmer and workers. The farmer (master or manager) process decomposes the problem in tasks and assigns them to the workers taking into account the dependencies between them. The workers (or slaves) typically perform most of the computational work by just executing those tasks.

The model has proved to be efficient in developing applications with different degrees of granularity of parallelism. This paradigm is particularly useful when the partitioning of the problem in tasks to be completed by the workers can be done easily and the dependencies between these tasks are low. In this point, the task farming paradigm is appropriate for processing in a Grid environment, since different degrees of granularity are available. Furthermore, if the worker tasks are completely independent from one another there is no need for synchronization between them that will very difficult.

A task farming application needs an interface to monitor the progress of the overall tasks and to steer individual task farming managers, to start, stop and track tasks.

An task farming application build on the current version of Globus Toolkit, GT4, that implements a service-oriented architecture, must be split to the farmer side as follows:

– The application controller is the component that launch the task farming requests into the Grid and expect to obtain some results. In the GT4 terms this part plays the client role. It can be written in any language, but it must respect the protocols used to contact the services. Moreover the client must be part of the virtual organization (shortly VO) in order to benefit from the services.

– The work splitter is the component that reacts to the controller requests by establishing the list of tasks to be performed and responds to the worker request to receive a task. In GT4 terms this part acts as an application-oriented specific service that is deployed into an active service container.

– The farm register keeps the evidence of the workers that are available to work. In GT4 terms this part is the virtual organization index, and it can be the MDS DefaultIndexService or another user-provided index service. The client can question the farm register in order to select or not the preferred workers (an automated scheduling mechanism is surely preferable).

– The work supervisor keeps track of the task status and notifies the client when some task is done. This is again an existing GT4 service, the ManagedJobFactoryService dealing with the Globus GRAM jobs.

The workers are receiving several tasks. Concrete objects (e.g. files) are received from the client and the splitter tell them what exactly should they do with these objects.

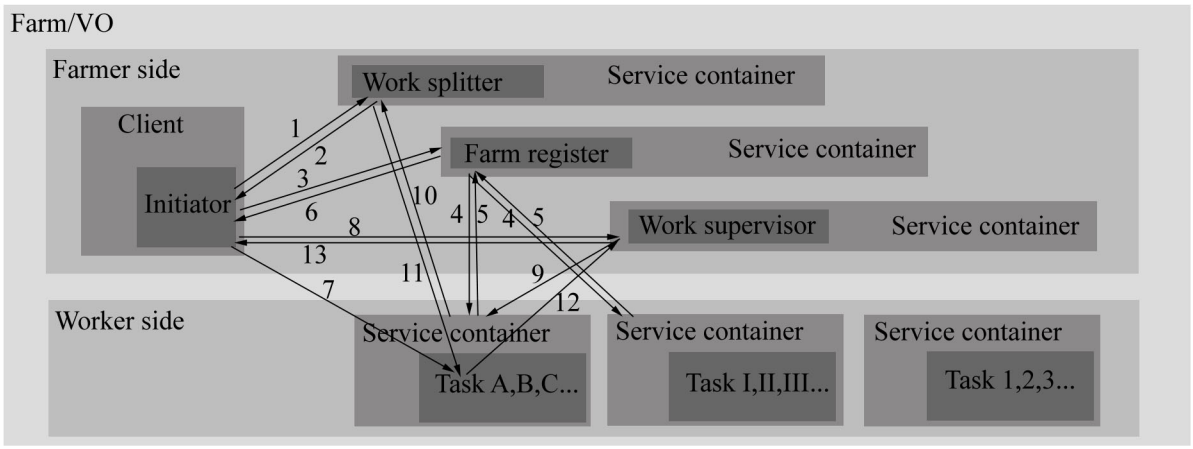


Figure 1: What happens in the farm

Figure 1 indicates the interaction between these different components:

a. the client contacts the splitter giving him enough information to be able to handle the splitting;
b. the splitter responds if it understood the message;
c. then the client contacts the VO index service to ask who many workers are available;
d. each worker (service container) registered in the index is request if it is available;
e. each reachable worker from the VO responds;
f. a list of available workers is provided;
g. the objects are transferred to the workers under the control of the work supervisor;
h. then tasks descriptions are send by the client to the supervisor;
i. the work supervisor send the request to work;
j. the workers are asking the work splitter about each task parameters;
k. the task parameters are send to the worker;
l. the work status is obtained by the supervisor;
m. the client is notified about the work status.

Each service container reside on a Grid node. The farmer's containers can be the same or different. The worker's containers are running on different nodes.

3 Case Study: An Application for Meteorology

Grid-enabled environments seem to have the potential to deal successfully with the vast amounts of data as produced in meteorology. There are currently several Grid projects dealing with meteorology applications. The list of application fields is long: aeronautical meteorology, agricultural meteorology, atmospheric research, environment, global climate observing, hydrology and water resources, marine meteorology, oceanography, public weather service, world climate, world weather watch. Most of Grid-based software prototypes are treating simulations on long time scale or planetary scale (e.g. numerical modelling for weather or climate research). A list of them is presented in [7]. We mention her only three significant projects.

Meteo-GRID is one of the three Grid application of EuroGRID [8]. The main goal of Meteo-GRID was the development of an application service provider solution that allows virtually anyone to run a high resolution numerical weather prediction model on demand.

One objective of the SIMDAT project, started in late 2004 [9], is to develop a virtual and consistent view of meteorological data distributed in real-time and archive databases and to provide a secure, reliable and efficient mechanism to collect, exchange and share these distributed data, in order to support research and operational activities of the meteorological community. Grid infrastructure will be enhanced to offer the distributed access to the different databases.

MetGRID [5] project, starting in 2006, will gather the major global weather forecasting centers of the world to share their global forecasts with each other as a multi-model ensemble.

Our application's approach is more simple. In the frame of our project we are interested to have a real-time image representing some meteorology data as public weather service.

As a starting example, we consider the particular case when we are interested to represent simple information like current temperature, pressure, wind speed, visibility, sky coverage and so on. A geographical region of interest should be pointed out. In what follows we considered the country were the conference take place.

Web services currently available are constantly updating the information of interest for this application. We depict one, the GlobalWeather Web Service [4], since a WSDL description of the service is available. Several towns from Romania are mentioned on the service list. At a proper request including the name of a town and its country, the service returns a response in XML format including several information about the latest registration for the following information: airport code, geographical coordinates, time of data registration, wind direction and speed, visibility, temperature, dew point, relative humidity, and pressure.

The task splitting method is simple. The map is divided into rectangles. Each rectangle is assigned to a worker. A list of towns and their coordinates on the map is available. If a town is in the worker rectangle, the worker request the appropriate information from the GlobalWeather Web Service.

Both parts of the application (farmer and worker) are written in Java. The client is a Java application presenting a GUI that drives the work and display the results. The SourceService takes the splitter role and it is implemented as a Grid service deployed into a GT4 container; it is initialized by the client. The workers are following the task description implemented a specific Java class. They are contacting the SourceService that will build an object describing the work to be done by the callers.

The map is given in SVG format, so that its content is easily modified.

The application GUI is a development of the code provided as an application example in [6]. Figure 2 presents on the left side the control panel with the services that are used by the client instance. In this particular case the VO Index (DefaultIndexService), the splitter (namely RenderSourceService) and the work supervisor (ManagedJobFactoryService) are deployed into the container running on the client machine (so that all the farmer pieces are lying on the same machine).

The Java classes representing the worker inputs are initially located on the source host in a specific directory and must be copied to the remote nodes.

After the parameter description, the grid is prepared, i.e. a request to the VO Index is send. A list with the available workers (node's names in this case) is presented. The user can select the nodes where the work will be send and the go buttons are starting the action: optionally, the file transfer, and then the tasks submissions.

The temperature was selected to be displayed. When the tasks are complete, a modified SVG map with changed colors according to the current temperature levels is rendered (a JPG file is generated too).

Figure 3 suggests how the work is split between the four workers. When larger number of rectangles than the worker number is used more, than the workers will receive more than one task to perform.

Although the example is very simple, the response time is improved by using several grid nodes instead only one. For the described example, the time for processing the Web service information has been reduced from 111 seconds for one node to 36 seconds for four nodes (mean value for 10 tries).

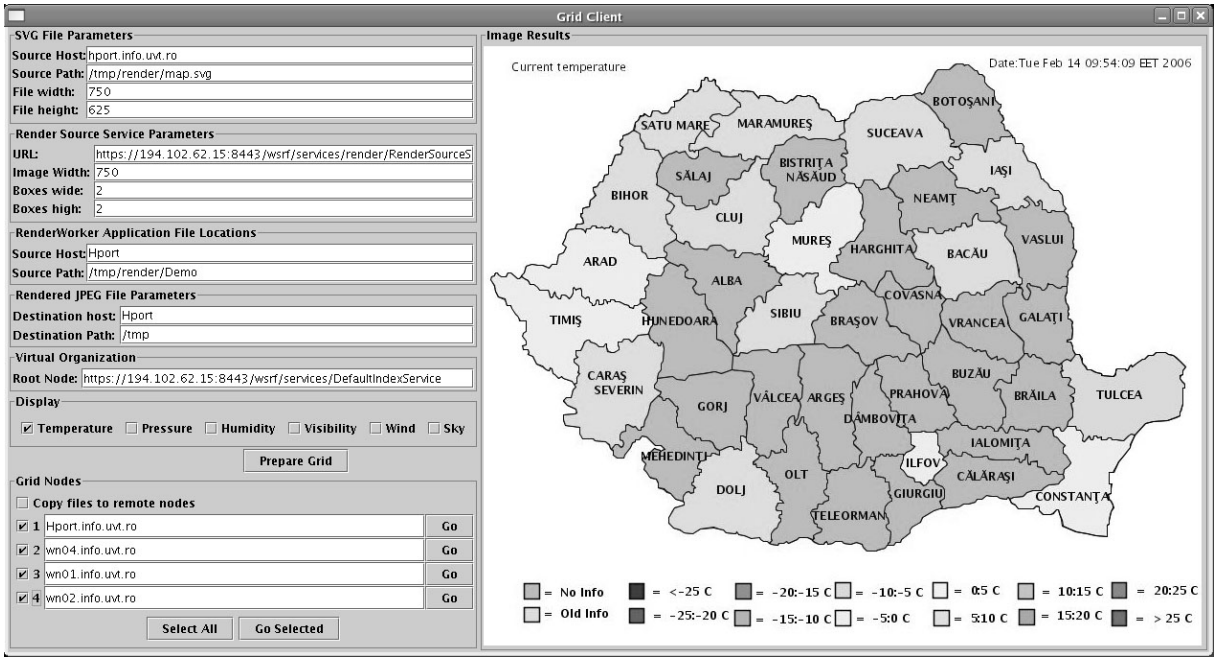


Figure 2: The user interface and the graphical result

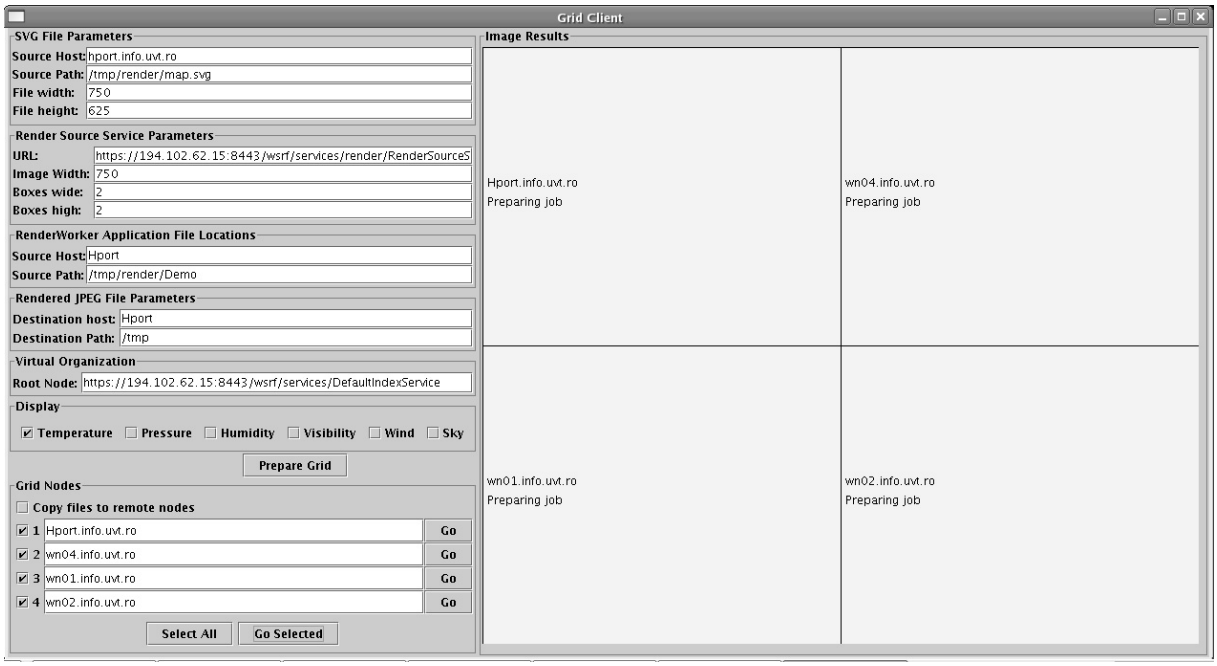


Figure 3: Intermediary image: task splitting between the Grid nodes

The application can be easily modified to periodically update the information and each intermediate resulting image can be further used, as frame in an animated sequence for example.

The complexity of the above described application is far from the complexity of existing software prototypes for real-time weather forecast or the real-time information. It is build only as a demo for the capabilities and benefits of the current Grid environments for meteorology applications.

4 Conclusions

The application described above is a simple example that proves that even in the case of a small number of tasks the task farming mode can be successfully implemented on Grids based on current technologies. It is a starting point for more complex applications in the field of Grid-based meteorology.

Acknowledgment

This work was supported by the Romanian project MedioGrid (CEEX-I03 19/07.10.2005).

References

[1] D. Abramson, J. Giddy, L. Kotler, High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?, in *Procs. IPDPS*, 14th Internat. Symposium on Parallel and Distributed Processing, Cancun, IEEE Computer Press, pp. 520-528, 2000.

[2] H. Casanova, M. Kim, J.S. Plank, J.J. Dongarra, Adaptive Scheduling for Task Farming with Grid Middleware, in *Internat. Journal of High Performance Computing Applications*, Vol. 13, No. 3, pp. 231-240, 1999.

[3] Globus Alliance, http://www.globus.org/.

[4] GlobalWeather Web Service, http://www.webservicex.net/globalweather.asmx

[5] G. Hoffman, Computing at DWD, in Procs. of Computing in Atmospheric Sciences Workshop, September 2005, Annency, France, http://www.cisl.ucar.edu/dir/CAS2K5.

[6] B. Jacob, M. Brown, K. Fukui, N. Trivedi, *Introduction to Grid Computing*, SG24-6778-00, Available at http://ibm.com/redbooks, 2005.

[7] R. Kaijun1, X. Nong, S. Junqiang, Z. Weimin1, W. Peng, The Research of a Semantic Architecture in Meteorology Grid Computing, in *Advanced Web and Network Technologies, and Applications*, Procs. APWeb 2006, eds. H.T. Shen et al., LNCS 3842, pp. 648 - 652, 2006.

[8] Meteo-GRID, http://www.eurogrid.org/wp2.html

[9] SIMDAT, Data Grids for Process and Product Development using Numerical Simulation and Knowledge Discovery, http://www.simdat.org

Dana Petcu
Institute e-Austria Timişoara
and
Western University of Timişoara
Computer Science Department
E-mail: petcu@info.uvt.ro

Cosmin Bonchiş
Institute e-Austria Timişoara
and
Western University of Timişoara
Computer Science Department

Maria Radu
Western University of Timişoara
Computer Science Department

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 376-380

The Importance of Parameters in Ant Systems

Camelia-Mihaela Pintea, Dan Dumitrescu

Abstract: In order to improve the solutions of *Traveling Salesman Problem, TSP* using ant system, we introduce a new correction rule in Ant Colony System. A parameter, similar to the temperature of simulated annealing, is dynamically modified. In this way it is possible to avoid stagnation after a number of iterations. It is a method to allow ants choosing other trails, in order to improve the solution. The benefit of the improvement is evident for big instances of Euclidean problems from *TSP* Library, *TSPLIB*.

Keywords: Ant System, Traveling Salesman Problem

1 Introduction

In the last years, ant systems, [16], becomes important optimization methods. They are combination of evolutionary computing and meta-heuristics. Similar to genetic algorithms, ant algorithms are inspired from natural process, simulating the path finding process. Ant systems are global optimizer methods, containing local optima avoidance techniques. Only the most representative members of the search space, thus the algorithm is a stochastic method, are examined.

Ants often find the shortest path between a food source and the nest of the colony without using visual information. In order to exchange information about which path should be followed, ants communicate with each other by means of a chemical substance called pheromone.

As ants move, a certain amount of pheromone is dropped on the ground, creating a pheromone trail. The more ants follow a given trail, the more attractive that trail becomes to be followed by other ants. This process involves a loop of positive feedback, in which the probability that an ant chooses a path is proportional to the number of ants that have already passed by that path.

Initially ant system has been applied for solving *Traveling Salesman Problem (TSP)* [16, 4]. A new ant-based algorithm, called *Q-Dynamic System, Q-DS* for solving *TSP* is proposed. Construct on the model of *Ant Colony System, ACS, Q-Dynamic System* introduce a new rule, the *q-rule*. The new correction rule dynamically modifies an ant colony system parameter. This is a way to improve the results of the algorithm, especially for big instances.

2 Ant Colony System for Traveling Salesman Problem

Ant algorithms are based on the real world phenomena that ants are able to find their way to a food source and back to their nest, using the shortest route.

See [4] where is described what about it happens when an ant comes across an obstacle and it has to decide the best route to take around the obstacle. Initially, there is equal probability as to which way the ant will turn in order to negotiate the obstacle.

If we assume that one route around the obstacle is shorter than the alternative route then the ants taking the shorter route will arrive at a point on the other side of the obstacle before the ants which take the longer route.

If we now consider other ants coming in the opposite direction, when they come across the same obstacle they are also faced with the same decision as to which way to turn. However, as ants walk they deposit a pheromone trail.

The ants that have already taken the shorter route will have laid a trail on this route so ants arriving at the obstacle from the other direction are more likely to follow that route as it has a deposit of pheromone. Over a period of time, the shortest route will have high levels of pheromone so that all ants are more likely to follow this route. There is positive feedback which reinforces that behavior so that the more ants that follow a particular route, the more desirable it becomes.

To convert this idea to a search mechanism for the *Traveling Salesman Problem* there are a number of factors to consider.

Initially the ants are placed randomly in the nodes of the graph. At iteration $t+1$ every ant moves to a new node and the parameters controlling the algorithm are updated.

Assuming that the *TSP* is represented as a fully connected graph, each edge is labeled by a trail intensity. Let $\tau_{ij}(t)$ represent the intensity of trail edge (i, j) at time t.

When an ant decides which node is the next move it does so with a probability that is based on the distance to that node and the amount of trail intensity on the connecting edge. The inverse of distance to the next node, is known as the *visibility*, η_{ij}. Visibility is defined as

$$\eta_{ij} = \frac{1}{d_{ij}},$$

where, d_{ij}, is the distance between nodes i and j.

At each time unit evaporation takes place. This is to stop the intensity trails increasing unbounded. The rate evaporation is denoted by ρ, and its value is between 0 and 1. In order to stop ants visiting the same node in the same tour a tabu list is maintained. This prevents ants visiting nodes they have previously visited.

To favor the selection of an edge that has a high pheromone value, τ, and high visibility value, η a function ${p^k}_{iu}$ is considered. ${J^k}_i$ are the unvisited neighbors of node i by ant k and $u \in {J^k}_i$. According to this function may be defined as

$${p^k}_{iu}(t) = \frac{[\tau_{iu}(t)][\eta_{iu}(t)]^\beta}{\Sigma_{o \in {J^k}_i}[\tau_{io}(t)][\eta_{io}(t)]^\beta}, \tag{1}$$

where β is a parameter used for tunning the relative importance of edge length in selecting the next node.

${p^k}_{iu}$ is the probability of choosing $j = u$ as the next node if $q > q_0$ (the current node is i). q is a random variable uniformly distributed over $[0,1]$ and q_0 is a parameter similar to the temperature in simulated annealing, $0 \leq q_0 \leq 1$. If $q \leq q_0$ the next node j is chosen as follows:

$$j = argmax_{u \in J_i^k}\{\tau_{iu}(t)[\eta_{iu}(t)]^\beta\}. \tag{2}$$

After each transition the trail intensity is updated using the correction rule:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_0. \tag{3}$$

Within *ACS* only the ant that generate the best tour is allowed to *globally* update the pheromone.

The global update rule is applied to the edges belonging to the *best tour*.

Let L^+ be the length of the best tour. The correction rule is:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t), \tag{4}$$

where $\Delta\tau_{ij}(t)$ is the inverse length of the best tour:

$$\Delta\tau_{ij}(t) = \frac{1}{L^+}. \tag{5}$$

The *Ant Colony System* is one of the most robust ant system, but being an heuristic, his results could be improved. For this purpose 2-opt. and 3-opt. algorithms are some of the well known and applied.

3 Q-Dynamic System

Traveling Salesman Problem (TSP) is solved with many heuristics, including ant system. For the same purpose it is introduced a new ant-based algorithm *Q-Dynamic System, (QDS)*. The *QDS* algorithm dynamically improve the solutions of *Ant Colony System*.

The new algorithm is modifying the value of q_0, a similar parameter to the temperature in simulated annealing. In general, $0 \leq q_0 \leq 1$. His value have a great influence in the choice of the new node.

As it was seen from the previous section, q is a random variable uniformly distributed over $[0,1]$ If $q \leq q_0$ the next node j is chosen as follows:

$$j = argmax_{u \in J_i^k}\{\tau_{iu}(t)[\eta_{iu}(t)]^\beta\}, \tag{6}$$

and otherwise,

$$p^k{}_{iu}(t) = \frac{[\tau_{iu}(t)][\eta_{iu}(t)]^\beta}{\Sigma_{o\in J^k{}_i}[\tau_{io}(t)][\eta_{io}(t)]^\beta}, \tag{7}$$

where $J^k{}_i$ is the set of unvisited neighbors of a node i used by the current ant, k. $p^k{}_{iu}$ is the probability of choosing $j = u$ as the next node if $q > q_0$ (the current node is i).

The correction rule, is applied after is randomly choosed q. The new rule, *q-rule*, is following.

$$q_0 = q_0 - c_0^1, \text{ If } q_0 < c_0^2 \text{ then } q_0 = v_0. \tag{8}$$

The values for c_0^1 and c_0^2 are in $[0,1)$. The v_0 could be any value in $[0,1]$, but it is best if it is chosen a value close to 1.

The *Q-DS* has the same structure as *Ant Colony System*.

begin
  set parameters, initialize pheromone trails
  repeat
    each ant is positioned on a starting node
    repeat
      each ant applies a state transition rule
       to incrementally build a solution
      and a local pheromone updating rule
    until all ants have built a complete solution
    a global pheromone updating rule is applied
  until end_condition
end

Table 1: Algorithmic skeleton for *Q-DS* algorithm

4 Tests and results

For numerical experiment we use problems from *TSPLIB* library [6]. *TSPLIB* provides optimal objective values for each of the problems. Several problems with Euclidean distances have been considered.

Until now is not developed a mathematical analysis for this model, which would give the optimal parameter in each situation.

We used as parameters, $\beta = 2$, $\rho = 0.1$, $q_0 = 0.5$ and $\tau_0 = (n \cdot L_{nn})^{-1}$.

L_{nn} is the result of *Nearest Neighbor, (NN)* algorithm. *NN* is perhaps the most natural heuristic for the *TSP*. In *NN* algorithm the rule is always to go next to the nearest as-yet-unvisited location. The corresponding tour traverses the nodes in the constructed order.

There is reiterated the new rule, called the *q-rule*.

$$q_0 = q_0 - c_0^1, \text{ If } q_0 < c_0^2 \text{ then } q_0 = v_0. \tag{9}$$

Good performances are obtain with c_0^1 and c_0^2 equal.

$$c_0^1 = c_0^2 = 0.1$$

Other good values are $c_0^1 = 0.05$ and $c_0^2 = 0.1$.

The value of $v_0 = 0.9$. Other good values are $v_0 \in [0.8, 0.99]$.

All the solutions are the average of five successively execution of the algorithm, for each problem. Termination criteria is given by the number of iteration $t_{max} = 100$, each on 10 trials.

It is very important the effect of the simultaneous presence of many ants, then each one contributes to the trail distribution, therefore, $m = 10$.

Problem	Optim	ACS with 2,3-opt	Q-DS
pr439	107217	**107217**	**107217**
pcb442	50778	**50778**	**50778**
rat783	8806	8815	**8812.2**
pcb1173	56892	56959.2	**56915**
d1291	50801	50820.4	**50801**

Table 2: The table is showing the optimal values of the *Ant Colony System* improved with 2 and 3-opt., and *Q-Dynamic System*. It is obvious that *Q-Dynamic System* has better results for big instances (*rat*783, *pcb*1173. *d*1291).

Good sets of edges will be followed by many ants and therefore will receive a great amount of trail. Bad sets of edges, chosen only to satisfy constraints, will be chosen only by few ants and therefore receive a small amount of trail.

Table 2 shows the *optimal* objective values for the problems, as given in [6] and the objective values returned by *Ant Colony System*, and *Q-Dynamic System*.

The tests show promising results for the *Q-Dynamic System*. The boldface solutions are the better solutions found for the algorithms in the specified conditions for chosen parameters. As it seen, the importance of the new rule is seen for big instances (e.g. *pr439* and *pcb442*).

It is very important to observe from Table 3, that the number of iteration is lower for the *Q-Dynamic System*, that implies a lower execution time for the new algorithm.

Problem	ACS iterations	Q-DS iterations
pr439	781	**771.30**
pcb442	783.10	**403.9**
rat783	1088.7	**986.2**
pcb1173	**550.7**	604.4
d1291	621.90	**432.5**

Table 3: The averages of iterations for *Ant Colony System* and *Q-Dynamic System* are shown for some euclidean instances. In general *Q-Dynamic System* has fewer iterations than *ACS* and implicitly a lower execution time for big instances.

An efficient combination with other techniques could improve the results. In general, tour improvement heuristics produce better solutions than tour constructive heuristic. It is more efficient to alternate an improvement heuristic with mutation of the last, or the best solution, rather than iteratively executing a tour improvement heuristic starting from solution generated randomly or by a constructive heuristic.

There are also non-Euclidean problems from *TSPLIB*, for which are known optimal objective values, that could be improved.

5 Conclusion

Q-Dynamic System, Q-DS is a new ant-based system. The new algorithm is the Ant Colony System improved with 2,3 opt. techniques and with a new correction rule for a valuable parameter.

The results of several tests shows that the new rule, called, the *q-rule* is in the benefit of *Traveling Salesman Problem* tours, especially for big instances. The execution time of *Q-DS* is shorter than for *Ant Colony System*. The parameters used for the *Q-DS* are still critical, as in all other ant systems.

References

[1] M.Dorigo, "Optimization, Learning and Natural Algorithms", Ph.D.Thesis, Politecnico di Milano, Italy, 1992

[2] M.Dorigo, L.Gambardella, "Ant Colonies for the Traveling Salesman Problem", *BioSystems*, Vol.43, pp.73–71, 1997.

[3] M.Dorigo, G.Caro, L.Gambardella, "Ant Algorithms for Discrete Optimization", *Artificial Life*, 5, 1999.

[4] M.Dorigo, V.Maniezzo, A.Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.26, pp.29-41, 1996

[5] C-M.Pintea, D.Dumitrescu, "Improving ant system using a local updating rule", *IEEE Proceedings of SYNASC, Symbolic and Numeric Algorithms for Scientific Computing*, 26-30, 2005

[6] G.Reinelt, "TSPLIB-A Traveling Salesman Problem Library", *ORSA Journal on Computing*, pp. 376-384, 1991.

[7] T.Stützle, H.H. Hoos, "$\mathcal{MAX}-\mathcal{MIN}$ Ant System", *Future Generation Computer Systems*, Vol. 16, pp.889-914, 2000.

C-M. Pintea, D.Dumitrescu
Babesş-Bolyai University of Cluj Napoca
Computer-Science Department
Kogalniceanu 1, Cluj-Napoca
E-mail: {cmpintea,ddumitr}@cs.ubbcluj.ro

This paper has been partially supported by the grant:
“Natural Computation. Applications in Bio/Nano Technology and Bioinformatics”
founded by Babeş-Bolyai University

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 381-385

On a Fuzzy Linguistic Approach to Solving Multiple Criteria Fractional Programming Problem

Bogdana Pop, Ioan Dziţac

Abstract: Mathematical model of multiple objective linear fractional programming problem is analyzed with respect to linguistic variables based solving methods. Two propositions are formulated related to choosing possibilities of aggregation coefficients for fractional criteria' membership functions. Computational results are developed in order to highlight theoretical remarks related to membership functions' for efficiency needed properties.

Keywords: fuzzy optimization, linguistic variable, multi-objective programming, linear fractional programming.

1 Introduction

In 2004 Rommelfanger ([6]) presented the advantages of fuzzy models in practical use. He points out that some interactive fuzzy solution algorithms provide the opportunity to solve mixed integer programming models as well. In 2002, Liu [4] presented a brief review on fuzzy programming models and classified them into three classes: expected value models, chance-constraint programming and dependent-chance programming. A general method to solve fuzzy programming models was also documented in Liu's paper.

Dutta, Tiwari and Rao [3] modified Luhandjula's linguistic approach ([5]) to obtain efficient solutions for MOLFPP (multiple objective linear fractional programming problem). In [8] some shortcomings are pointed out and a correct proof of Dutta's et al. main theorem is given. Moreover, it is noticed that the method presented in [3] only works efficiently if some quite restrictive hypotheses are satisfied. Chakraborty and Gupta [2] described a new fuzzy method to solving MOLFPP improving the complexity of computations by defining fuzzy goals for a deterministic MOLFPP.

In [2] Carlsson et al. considered a mathematical programming problem in which the functional relationship between the decision variables and the objective function is not completely known and built a knowledge-base which consists of a block of fuzzy if-then rules, where the antecedent part of the rules contains some linguistic values of the decision variables, and the consequence part is a linear combination of the crisp values of the decision variables.

The concept of linguistic variable was introduced by Zadeh [4] to provide a means of approximate characterization of phenomena that are too complex or too vague (not well)-defined to be described in conventional quantitative terms.

Mathematical model of MOLFPP is presented in Section2. Comments on linguistic variables based solving methods are formulated in Section 3. Some computational results are developed in Section 4 in order to highlight theoretical remarks which were made before. Brief summary and conclusions are inserted in Section 5.

2 Mathematical model of MOLFPP

Consider the multiple objective linear fractional programming problem (MOLFPP):

$$\text{"max"}\left\{z(x)=\left(\frac{N_1(x)}{D_1(x)},\frac{N_2(x)}{D_2(x)},...,\frac{N_p(x)}{D_p(x)}\right)\mid x\in X\right\} \tag{1}$$

where

(i) $X=\{x\in R^n \mid Ax\leq b, x\geq 0\}$ is a convex and bounded set,

(ii) A is an $m\times n$ constraint matrix, x is an n-dimensional vector of decision variable and $b\in R^m$,

(iii) $p\geq 2$,

(iv) $N_i(x)=\left(c^i\right)'x+d_i,\ D_i(x)=\left(e^i\right)'x+f_i, \forall i=\overline{1,p}$,

(v) $c^i, e^i\in R^n, d_i, f_i\in R, \forall i=\overline{1,p}$,

(vi) $\left(e^i\right)'x+f_i>0, \forall i=\overline{1,p}, \forall x\in X$.

The term "max" being used in Problem (1) is for finding all weakly efficient and strongly efficient solutions in a maximization sense in terms of classic definitions [7].

To solve a multiple objective programming problem means to find a compromise solution. In this case, the fuzzy set theory proposes appropriate modeling tools for handling compromises. The goal of obtaining better solutions is connected to the goal of choosing appropriate fuzzy aggregation model depending on the application's specific nature.

3 On linguistic variables based solving methods

Imprecise aspirations of the decision-maker can be represented by structured linguistic variable. The concept of (Z,ε)-proximity will be used in the larger framework of the linguistic variables domain which leads with following membership functions:

$$C_j^{N_i}(x)=\begin{cases} 0 \quad , & \text{if} \quad N_i(x)<p_i^j \\ \dfrac{N_i(x)-p_i^j}{N_i^0-p_i^j}, & \text{if} \quad p_i^j\le N_i(x)\le N_i^0 \\ 0 \quad , & \text{if} \quad N_i(x)>N_i^0 \end{cases} ,\forall i=\overline{1,p}, \tag{2}$$

$$C_j^{D_i}(x)=\begin{cases} 0 \quad , & \text{if} \quad D_i(x)>s_i^j \\ \dfrac{s_i^j-D_i(x)}{s_i^j-D_i^0}, & \text{if} \quad D_i^0\le D_i(x)\le s_i^j \\ 0 \quad , & \text{if} \quad D_i(x)<D_i^0 \end{cases} ,\forall i=\overline{1,p} \tag{3}$$

or

$$C_j^{z_i}(x)=\begin{cases} 0 \quad , & \text{if} \quad z_i(x)<r_i^j \\ \dfrac{z_i(x)-r_i^j}{z_i^0-r_i^j}, & \text{if} \quad r_i^j\le z_i(x)\le z_i^0 \\ 0 \quad , & \text{if} \quad z_i(x)>z_i^0 \end{cases} ,\forall i=\overline{1,p}. \tag{4}$$

where N_i^0, D_i^0 and z_i^0 $\left(\forall i=\overline{1,p}\right)$ represent the maximal value of nominator $N_i(x)$, the minimal value of denominator $D_i(x)$ and the maximal value of linear fractional functions $z_i(x)$ on the set X, while p_i^j, s_i^j, r_i^j $(j=1,2,3)$ are the thresholds beginning with which values $N_i(x)$, $D_i(x)$ and $z_i(x)$ are (quite close, close, very close) acceptable.

When membership functions (2)-(3) are used an aggregation of them are made later in order to obtain a membership function for objective functions. These membership functions are better to be used than membership functions (4) because of linearity. Despite of this, an obtained membership function could loose essential properties of corresponding objective function if the aggregation operator is not well selected.

The aim of this paper is to analyze some aspects which must be taken into consideration when coefficients' selection in a linear combination of linear membership function is made. Next we will focus on one single objective function. Consequently, above i indexes will not be necessary to be used and we will eliminate them in order to avoid complicated formulas. Thresholds' j indexes will be also eliminated. Next we work with following membership functions.

$$C^N(x)=\begin{cases} 0 \quad , & \text{if} \quad N(x)<p \\ \dfrac{N(x)-p}{N^0-p}, & \text{if} \quad p\le N(x)\le N^0 \\ 0 \quad , & \text{if} \quad N(x)>N^0 \end{cases} ,$$

for fuzzyfing nominator's maximization goal and

$$C^D(x)=\begin{cases} 0 \quad , & \text{if} \quad D(x)>s \\ \dfrac{s-D(x)}{s-D^0}, & \text{if} \quad D^0\le D(x)\le s \\ 0 \quad , & \text{if} \quad D(x)<D^0 \end{cases}$$

for fuzzyfing denominator's minimization goal.

When a linear aggregation $\mu(x)=wC^N(x)+w'C^D(x)$ is made new membership function $\mu(x)$ would be better to verify hypothesis (5) in order to retain all essential properties of initial linear fractional function $z(x)$ from the optimization point of view.

$$\forall x^1, x^2 \in X, z\left(x^1\right) > z\left(x^2\right) \text{ then } \mu\left(x^1\right) > \mu\left(x^2\right) \tag{5}$$

In literature hypothesis (5) is stated to be verified or it is replaced by equivalent hypotheses. In [8] it is proved that coefficients w and w' must verified $k\underline{A} < w'/w < k\overline{A}$ where

$$\overline{A} = \min\left\{ \frac{N\left(x^1\right) - N\left(x^2\right)}{D\left(x^1\right) - D\left(x^2\right)} \mid D\left(x^1\right) < D\left(x^2\right), \frac{N\left(x^1\right)}{D\left(x^1\right)} < \frac{N\left(x^2\right)}{D\left(x^2\right)}, x^1, x^2 \in X \right\}$$

and

$$\underline{A} = \max\left\{ \frac{N\left(x^1\right) - N\left(x^2\right)}{D\left(x^1\right) - D\left(x^2\right)} \mid D\left(x^1\right) > D\left(x^2\right), \frac{N\left(x^1\right)}{D\left(x^1\right)} < \frac{N\left(x^2\right)}{D\left(x^2\right)}, x^1, x^2 \in X \right\}.$$

Using transformation $y = x^1 - x^2, x^1, x^2 \in X$ following equivalent formulas (6) to calculate $\overline{A}$ and $\underline{A}$ are obtained.

$$\overline{A} = \min\left\{ \frac{c'y}{e'y} \mid e'y < 0, \frac{c'y}{d'y} > z\left(x^2\right) \right\}, \quad \underline{A} = \max\left\{ \frac{c'y}{e'y} \mid e'y > 0, \frac{c'y}{e'y} < z\left(x^2\right) \right\}. \tag{6}$$

Because of propositions below we can conclude that hypothesis (5) and its equivalent forms could give an empty range for w'/w.

Proposition 1. *Hypothesis (5) can be verified by $z(x)$ and $\mu(x)$ if and only if*

$$\frac{\partial z}{\partial x_i} \cdot \frac{\partial \mu}{\partial x_i} \geq 0 \text{ for each } i = 1, 2, ..., n.$$

Proposition 2. *A non-empty range for w'/w to verifying hypothesis (5) exists if and only if*

$$h_i(x) = \left(c_i e' - e_i c'\right) x + c_i f - e_i d, x \in X, i = 1, 2, ..., n$$

doesn't change its sign over X, for each index i.

Proposition (1) is a consequence of monotony from the $1-$dimensional case. In fact, hypothesis (5) states monotony along any direction. Proposition (2) is a consequence of computing partial derivatives. Derivatives of linear function μ are constant and the sign of derivatives of linear fractional function z is stated by h_i.

$$\frac{\partial}{\partial x_i}\left(\frac{c'x + d}{e'x + f}\right) = \frac{\left(c_i e' - e_i c'\right) x + c_i f - e_i d}{\left(e'x + f\right)^2}$$

$$\frac{\partial}{\partial x_i}\left[\alpha\left(c'x + d\right) + \beta\left(e'x + f\right)\right] = \alpha c_i + \beta e_i$$

Coefficients α and β involve coefficients w and w', thresholds p and s and marginal solutions N^0 and D^0 of nominator and denominator respectively.

Returning to MOLFPP we have to conclude that any solving method can be improved by choosing proper membership functions for each criterion. An $n-$dimensional MOLFPP can be considered as being $2n-$dimensional MOLPP (multiple objective linear programming problem) if all fractional objective functions in MOLFPP come from $2-$dimensional models. Otherwise, for fractional objective functions it is better to consider linguistic variable based membership functions which can retain their more essential properties.

4 Computational results

In this section, the example considered in [3] and [8] is discussed.

Example 3.

$$\max\left(z_1(x) = \frac{x_1 + x_2 - 1}{-x_1 + 2x_2 + 7}, z_2(x) = \frac{2x_1 + x_2 - 2}{x_2 + 4}\right) \tag{7}$$

subject to

$$\begin{aligned} &-x_1 + 3x_2 \leq 0, \\ &x_1 \leq 6, \\ &x_1, x_2 \geq 0. \end{aligned} \tag{8}$$

The single efficient point of Problem (7)-(8) is $x^{opt} = (6,0)$. Both objective functions reach in this point their optimum, independently one from another, on the same feasible region. Marginal solution of functions z_1, z_2 and of their nominators and denominators, marginal points of the feasible set X and thresholds p_1, p_2, s_1, s_2 are described in the table below.

Marginal points	f_1	N_1	p_1	D_1	s_1	f_2	N_2	p_1	D_2	s_2
$(0,0)$	-0.142	-1	4	7	3	-0.5	-2	8	4	7
$(6,0)$	5	5	4	1	3	2.5	10	8	4	7
$(6,1)$	1.6	8	4	5	3	2	12	8	6	7

Optimizing (6) following results are obtained.

i	$\overline{A}_i$	$\underline{A}_i$	Range for w_i'/w_i	Remarks
1	1.46189	4.98327	$[3.334, 1]$	$\emptyset$
2	1.47488	2.49800	$[1.875, 1]$	$\emptyset$

Even the range of w'/w is empty selecting different values for w', w efficient or non-efficient solution can be obtained as it can be seen below.

w_1	w_1'	w_1'/w_1	w_2	w_2'	w_2'/w_2	Remarks
0.035	0.465	13.285	0.475	0.025	5.263	Efficient solution
0.991	0.007	7.063	0.001	0.001	1	Non-efficient solution

Information from the next table states that hypothesis (5) couldn't be verified for any set of aggregation coefficients w, w'.

i	$h_i^1(x)$	Remarks
1	$3x_2 + 6$	$\geq 0, \forall x \in X$
2	$-3x_1 + 9$	≤ 0 for $x_1 \geq 3$, ≥ 0 for $x_1 \leq 3$

5 Conclusions

Advantages of fuzzy models in practical use and general methods to solve fuzzy programming models were synthesized so far ([2, 4, 6]). Fuzzy approaches to solve deterministic problems were developed in recent literature.

We have addressed linguistic variable based method for solving linear fractional programming problems. Two propositions were formulated related to choosing possibilities of aggregation coefficients for fractional criteria' membership functions.

Computational results were developed in order to highlight theoretical remarks related to membership functions' for efficiency needed properties.

We have concluded that any solving method can be improved by choosing proper membership functions for each criterion in order to retain more essential properties of models. Also, an $n-$dimensional fractional problem it is better to be considered as being $2n-$dimensional linear problem if all fractional objective functions come from $2-$dimensional models.

References

[1] C. Carlsson, R. Fuller, "Multiobjective Linguistic Optimization", *Fuzzy Sets and Systems*, Vol. 115, pp. 5-10, 2000.

[2] M. Chakraborty, Sandipan Gupta, "Fuzzy Mathematical Programming for Multi Objective Linear Fractional Programming Problem", *Fuzzy Sets and Systems*, Vol. 125, pp. 335-342, 2002.

[3] D. Dutta, R.N. Tiwari, J.R. Rao, "Multiple Objective Linear Fractional Programming Problem - A Fuzzy Set Theoretic Approach", *Fuzzy Sets and Systems*, Vol. 52, pp 39-45, 1992.

[4] B. Liu, "Toward Fuzzy Optimization without Mathematical Ambiguity", *Fuzzy Optimization and Decision Making*, Vol. 1, pp. 43-63, 2002.

[5] M.K. Luhandjula, "Fuzzy approaches for multiple objective linear fractional optimization", *Fuzzy Sets and Systems*, Vol. 13, pp. 11-23, 1984.

[6] H.J. Rommelfanger, "The Advantages of Fuzzy Optimization Models in Practical Use", *Fuzzy Optimization and Decision Making*, Vol. 3, pp. 295-309, 2004.

[7] I.M. Stancu-Minasian, *Fractional Programming: Theory, Methods and Applications*, Kluwer Academic Publishers, Dordrecht, 1997.

[8] I.M. Stancu-Minasian, Bogdana Pop, "On a Fuzzy Set Approach to Solve Multiple Objective Linear Fractional Programming Problem", *Fuzzy Sets and Systems*, Vol. 134, pp. 397-405, 2003.

[9] L.A. Zadeh, "The Concept of Linguistic Variable and its Application to Approximate Reasoning", *Inform. Sci.*, Vol. 8, pp. 199-244, 1975.

Bogdana Pop
"Transilvania" University of Braşov
Computer Science Department
Address: Iuliu Maniu 50, 500091 Braşov, Romania
E-mail: bgdnpop@gmail.com

Ioan Dziţac
Agora University
Deptartment of Business Informatics
Address: 8, Piaţa Tineretului St., 410526, Oradea, Romania
E-mail: idzitac@univagora.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 386-392

An Off-line Electronic Cash System with Multiple Banks

Constantin Popescu, Horea Oros

Abstract: Secure and efficient electronic payment systems are significant for electronic commerce. Fangguo [9] proposed a fair electronic cash system with multiple banks based on a group signature scheme of Camenisch [5] and the group blind signature scheme of Lysyanskaya [11]. Ateniese proved in [1] that these group signature schemes does not satisfy the property of coalition-resistance. In this paper we extend the electronic cash system of Fangguo by using a secure coalition-resistant group blind signature scheme. The main benefits of our off-line electronic cash system, compared to the scheme of Fangguo, relate to the underlying group signature scheme's improved efficiency and provable security.

Keywords: Cryptography, electronic cash system, group blind signatures.

1 Introduction

The first electronic cash system was suggested by Chaum [7] in 1982. Chaum used the technique of blind signatures in order to guarantee the privacy of users. Von Solms and Naccache showed in [18] that anonymity could be used for blackmailing or money laundering by criminals without revealing their identities.

Fair electronic cash systems have been suggested independently by Brickell, Gemmel and Kravitz [3] and Stadler, Piveteau and Camenisch [16] as a solution to prevent blackmailing and money laundering. The efficiency and the security of the scheme in [3] have been later improved in [10]. Also, various fair electronic cash systems using group signature schemes have been proposed in [6], [12], [14], [17]. Traore [17] proposed a solution that combine a group signature scheme and a blind signature scheme in order to design a fair off-line electronic cash. Recently, Qiu et al. [15] presented a new electronic cash system using a combination of a group signature scheme and a blind signature scheme. Canard and Traore [6] and Choi, Zhang and Kim [8] suggested that the Qiu's system does not provide the anonymity of the customers.

In the above electronic cash systems, the electronic coins are issued by one bank. However, in practice it is more convenient to use electronic coins issued by multiple banks from a country. Each of these banks can issue electronic coins of its own. These banks form a group under control of an authorized party such as the Central Bank of the country.

Fangguo [9] proposed a fair electronic cash system with multiple banks based on a group signature scheme of Camenisch [5] and the group blind signature scheme of Lysyanskaya [11].

In their payment system, they used a group signature scheme of Camenisch and Stadler [5] for large groups which is not secure. Ateniese proved in [1] that this group signature scheme does not satisfy the property of coalition-resistance.

In this paper we extend the electronic cash system of Fangguo [9] by using a secure coalition-resistant group blind signature scheme [13].

The remainder of this paper is organized as follows. In the next section, we present a model for off-line electronic cash systems with multiple banks. Then, we present our off-line electronic payment system in Section 3. Furthermore, we discuss some some aspects of security and efficiency in Section 4. Finally, Section 5 concludes the work of this paper.

2 The Model for Off-line Electronic Cash Systems with Multiple Banks

In the model of Fangguo [9] there are many banks. These banks form a group under control of the Central Bank. The Central Bank plays the role of the group manager in a group signature scheme. In an off-line electronic cash system with multiple banks [9], the following parties are involved: the Central Bank B, many local banks $B_1, B_2, ..., B_t$, a trusted third party T, some customers $C_1, C_2, ..., C_z$ and some merchants M. The basic model for off-line electronic cash system with multiple banks is presented in Figure 1.

The model includes the following protocols:

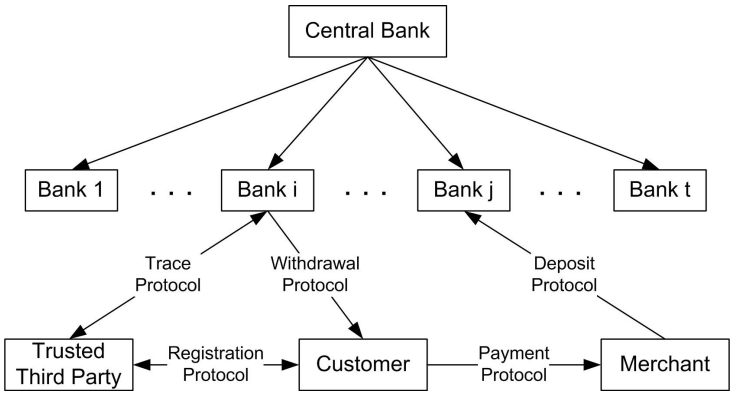


Figure 1: The model for off-line electronic cash system with multiple banks

a. Registration Protocol. The customer C establishes the relationship of his identity to the trusted third party. So, the trusted third party can revoke the customer anonymity when is necessary. At the same time the customer gets a certificate issued by the trusted third party that proves his registration.

b. Open account Protocol. The customer C_i opens his account in his bank B_i.

c. Withdrawal Protocol. The customer C_i withdraws an electronic coin from his bank B_i.

d. Payment Protocol. The customer C_i purchases goods in the merchant M and pays the coin to the merchant M.

e. Deposit Protocol. The merchant M deposits the received coin to his account in his bank B_j.

f. Trace Protocol. When illegal acts like blackmailing, money laundering and illegal purchases are disclosed, the bank B_j sends the received coin to the Central Bank. Then the Central Bank finds the bank B_i and the customer C_i can be traced with the help of the trusted third party.

3 The Proposed Off-line Electronic Cash System with Multiple Banks

In the proposed electronic cash system all banks form a group with the Central Bank as group manager and all customers also form a group with a trusted third party as group manager. Each bank can issue coins using the technique of group blind signature scheme.

3.1 Registration Protocol

The Central Bank uses the group signature scheme of Ateniese, Camenisch, Joye and Tsudik [2] in order to select the following parameters:

a. Let k, l_p and $\varepsilon > 1$ be security parameters and let $\lambda_1, \lambda_2, \gamma_1, \gamma_2$ denote lengths satisfying $\lambda_1 > \varepsilon(\lambda_2 + k) + 2, \lambda_2 > 4l_p, \gamma_1 > \varepsilon(\gamma_2 + k) + 2$ and $\gamma_2 > \lambda_1 + 2$. Define the integral ranges $\Lambda = [2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}]$ and $\Gamma = [2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}]$.

b. Select random secret l_p-bit primes p', q' such that $p = 2p' + 1$ and $q = 2q' + 1$ are prime. Set the modulus $n = pq$. It is a good habit to restrict operation to the subgroup of quadratic residues modulo n, i.e., the cyclic subgroup $QR(n)$ generated by an element of order $p'q'$. This is because the order $p'q'$ of $QR(n)$ has no small factors.

c. Choose random elements $a, a_0, g, h \in QR(n)$ of order $p'q'$.

d. Choose a random secret element $x \in \mathbb{Z}^*_{p'q'}$ and set $y = g^x \bmod n$.

e. Finally, let H be a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^k$.

f. The group public key is $P = (n, a, a_0, H, y, g, h, l_G, \lambda_1, \lambda_2, \gamma_1, \gamma_2)$.

g. The corresponding secret key is $S = (p', q', x)$. This is the Central Bank's secret key.

The trusted third party executes the same steps as the Central Bank to setup parameters with the following modifications:

a. Choose random elements $a', a_0, g', h \in QR(n)$ of order $p'q'$.

b. Choose a random secret element $x' \in \mathbb{Z}^*_{p'q'}$ and set $y' = g'^{x'} \bmod n$.

c. The group public key is $P' = (n, a', a_0, H, y', g', h, l_G, \lambda_1, \lambda_2, \gamma_1, \gamma_2)$.

d. The corresponding secret key is $S' = (p', q', x')$. This is the secret key of the trusted third party.

A bank B_i joins the group of banks and gets its membership certificate. To obtain its membership certificate, each bank B_i must perform the following protocol with the Central Bank:

a. Generates a secret key $x_i \in \Lambda$. The corresponding public key is $y_i = a^{x_i} \bmod n$. The bank B_i also proves to the Central Bank that the discrete logarithm of y_i with respect to base a lies in the interval Λ (see [4]).

b. The Central Bank sends to the bank B_i the new membership certificate (A_i, e_i), where e_i is a random prime chosen by the Central Bank such that $e_i \in \Gamma$ and A_i has been computed by the Central Bank as $A_i = (y_i a_0)^{1/e_i} \bmod n$.

c. The Central Bank creates a new entry in the membership table and stores (A_i, e_i) in the new entry.

When a customer C_i registers in the trusted third party, he gets a membership certificate and become a legal member of the customer group. To obtain his membership certificate, each customer C_i must perform the following protocol with the trusted third party:

a. Generates a secret key $x''_i \in \Lambda$. The corresponding public key is $y'_i = a'^{x''_i} \bmod n$. The customer C_i also proves to the trusted third party that the discrete logarithm of y'_i with respect to base a' lies in the interval Λ (see [4]).

b. The trusted third party sends to the customer C_i the new membership certificate (A'_i, e'_i), where e'_i is a random prime chosen by the trusted third party such that $e'_i \in \Gamma$ and A'_i has been computed by the trusted third party as $A'_i = (y'_i a_0)^{1/e'_i} \bmod n$.

c. The trusted third party creates a new entry in the membership table and stores (A'_i, e'_i) in the new entry.

3.2 The Withdrawal Protocol

The withdrawal protocol involves the customers and the banks. A customer has his account number in his bank B_i. When the customer wants to withdraw an electronic coin, he first performs a protocol with B_i to authenticate his identity and his account number. If succeed, the customer generates an electronic coin m and gets the group blind signature of B_i on m by performing the following protocol:

The bank B_i does the following:

a. Computes:

$$\tilde{A} = A_i y^{x_i} \,(\bmod n)\,,\ \tilde{B} = g^{x_i} \,(\bmod n)\,,\ \tilde{D} = g^{e_i} h^{x_i} \,(\bmod n) \tag{1}$$

b. Chooses random values $\tilde{r_1} \in \pm\{0,1\}^{\varepsilon(\gamma_2+k)}$, $\tilde{r_2} \in \pm\{0,1\}^{\varepsilon(\lambda_2+k)}$, $\tilde{r_3} \in \pm\{0,1\}^{\varepsilon(\gamma_1+2l_p+k+1)}$, $\tilde{r_4} \in \pm\{0,1\}^{\varepsilon(2l_p+k)}$ and computes:

$$\tilde{t_1} \;=\; \tilde{A}^{\tilde{r_1}} / \left(a^{\tilde{r_2}} y^{\tilde{r_3}}\right),\ \tilde{t_2} = \tilde{B}^{\tilde{r_1}} / g^{\tilde{r_3}},\ \tilde{t_3} = g^{\tilde{r_4}}, \tilde{t_4} = g^{\tilde{r_1}} h^{\tilde{r_4}} \tag{2}$$

c. Sends $(\tilde{A}, \tilde{B}, \tilde{D}, \tilde{t_1}, \tilde{t_2}, \tilde{t_3}, \tilde{t_4})$ to the customer.

In turn, the customer does the following:

a. Chooses $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \delta \in_R \{0,1\}^{\varepsilon(l_p+k)}$ and computes:

$$t_1 = a_0^{\delta}\tilde{t_1}\tilde{A}^{\alpha_1-\delta 2^{\gamma_1}}/(a^{\alpha_2-\delta 2^{\lambda_1}}y^{\alpha_3}),\ t_2 = \tilde{t_2}\tilde{B}^{\alpha_1-\delta 2^{\gamma_1}}/g^{\alpha_3} \quad (3)$$

$$t_3 = \tilde{t_3}\tilde{B}^{\delta}g^{\alpha_4},\ t_4 = \tilde{t_4}\tilde{D}^{\delta}g^{\alpha_1}h^{\alpha_4} \quad (4)$$

b. Computes:

$$c = H(m\|g\|h\|y\|a_0\|a\|\tilde{A}\|\tilde{B}\|\tilde{D}\|t_1\|t_2\|t_3\|t_4) \quad (5)$$

$$\tilde{c} = c-\delta \quad (6)$$

c. Sends $\tilde{c}$ to the signer.

The bank B_i does the following:

a. Computes:

$$\tilde{s_1} = \tilde{r_1}-\tilde{c}(e_i-2^{\gamma_1}),\ \tilde{s_2} = \tilde{r_2}-\tilde{c}(x_i-2^{\lambda_1}) \quad (7)$$

$$\tilde{s_3} = \tilde{r_3}-\tilde{c}e_ix_i,\ \tilde{s_4} = \tilde{r_4}-\tilde{c}x_i \quad (8)$$

b. Sends $(\tilde{s_1},\tilde{s_2},\tilde{s_3},\tilde{s_4})$ to the user.

The customer does the following:

a. Computes:

$$s_1 = \tilde{s_1}+\alpha_1,\ s_2 = \tilde{s_2}+\alpha_2\ s_3 = \tilde{s_3}+\alpha_3 \quad (9)$$

$$s_4 = \tilde{s_4}+\alpha_4,\ A = \tilde{A}^{H(c\|s_1\|s_2\|s_3\|s_4)} \bmod n \quad (10)$$

$$B = \tilde{B}^{H(c\|s_1\|s_2\|s_3\|s_4)} \bmod n, D = \tilde{D}^{H(c\|s_1\|s_2\|s_3\|s_4\|A\|B)} \bmod n \quad (11)$$

b. The resulting group blind signature of a message m is $(c,s_1,s_2,s_3,s_4,A,B,D)$.

3.3 The Payment Protocol

The payment protocol involves the customers and the merchant.

a. The customer sends to the merchant the group blind signature $\sigma = (c,s_1,s_2, s_3,s_4,A,B,D)$ of the message m.

b. The merchant first verifies the validity of the group blind signature $\sigma = (c,s_1,s_2, s_3,s_4,A,B,D)$ with the public key P as follows:

(a) Computes:

$$b_1 = 1/H(c\|s_1\|s_2\|s_3\|s_4) \quad (12)$$

$$b_2 = 1/H(c\|s_1\|s_2\|s_3\|s_4\|A\|B) \quad (13)$$

$$t_1' = a_0^c A^{b_1(s_1-c2^{\gamma_1})}/(a^{s_2-c2^{\lambda_1}}y^{s_3}) \bmod n \quad (14)$$

$$t_2' = B^{b_1(s_1-c2^{\gamma_1})}/g^{s_3} \bmod n \quad (15)$$

$$t_3' = B^{cb_1}g^{s_4} \bmod n \quad (16)$$

$$t_4' = D^{cb_2}g^{s_1-c2^{\gamma_1}}h^{s_4} \bmod n \quad (17)$$

$$c' = H(m\|g\|h\|y\|a_0\|a\|A^{b_1}\|B^{b_1}\|D^{b_2}\|t_1'\|t_2'\|t_3'\|t_4') \quad (18)$$

(b) Accept the group blind signature if and only if:

$$c = c' \quad (19)$$

$$s_1 \in \pm\{0,1\}^{\varepsilon(\gamma_2+k)+1}, \qquad s_2 \in \pm\{0,1\}^{\varepsilon(\lambda_2+k)+1} \quad (20)$$

$$s_3 \in \pm\{0,1\}^{\varepsilon(\lambda_1+2l_p+k+1)+1}, \qquad s_4 \in \pm\{0,1\}^{\varepsilon(2l_p+k)+1} \quad (21)$$

c. The customer computes $m' = H(c\|s_1\|s_2\|s_3\|s_4\|A\|B\|D)$ and signs m' using the group signature scheme proposed by Ateniese, Camenisch, Joye and Tsudik [2]:

(a) Chooses a random integer $w' \in \{0,1\}^{2l_p}$ and computes:

$$T_1 = A_i' y'^{w'} \pmod{n}, T_2 = g'^{w'} \pmod{n}, T_3 = g'^{e_i'} h'^{w'} \pmod{n} \tag{22}$$

(b) Randomly chooses:

$$r_1 \in \pm\{0,1\}^{\varepsilon(\gamma_2+k)}, \qquad r_2 \in \pm\{0,1\}^{\varepsilon(\lambda_2+k)} \tag{23}$$

$$r_3 \in \pm\{0,1\}^{\varepsilon(\gamma_1+l_p+k+1)}, \qquad r_4 \in \pm\{0,1\}^{\varepsilon(2l_p+k)} \tag{24}$$

(c) Computes:

$$d_1 = T_1^{r_1}/(a'^{r_2} y'^{r_3}), d_2 = T_2^{r_1}/g'^{r_3}, d_3 = g'^{r_4}, d_4 = g'^{r_1} h^{r_4} \tag{25}$$

(d) Computes:

$$c_1 = H(m'\|g'\|h\|y'\|a_0\|a'\|T_1\|T_2\|T_3\|d_1\|d_2\|d_3\|d_4) \tag{26}$$

$$s_1' = r_1 - c_1(e_i' - 2^{\gamma_1}),\ s_2' = r_2 - c_1(x_i'' - 2^{\lambda_1}) \tag{27}$$

$$s_3' = r_3 - c_1 e_i' w',\ s_4' = r_4 - c_1 w' \tag{28}$$

(e) The resulting group signature of a message m' is $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$.

d. The customer sends the merchant the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ of the message m'.

e. The merchant verifies the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ of the message m' with public key P' as follows:

(a) Computes:

$$d_1' = a_0^{c_1} T_1^{s_1' - c_1 2^{\gamma_1}} / (a'^{s_2' - c_1 2^{\lambda_1}} y'^{s_3'}) \bmod n \tag{29}$$

$$d_2' = T_2^{s_1' - c_1 2^{\gamma_1}} / g'^{s_3'} \bmod n \tag{30}$$

$$d_3' = T_2^{c_1} g'^{s_4'} \bmod n \tag{31}$$

$$d_4' = T_3^{c_1} g'^{s_1' - c_1 2^{\gamma_1}} h^{s_4'} \bmod n \tag{32}$$

$$c_1' = H(m'\|g'\|h\|y'\|a_0\|a'\|T_1\|T_2\|T_3\|d_1'\|d_2'\|d_3'\|d_4') \tag{33}$$

(b) Accept the group signature if and only if:

$$c_1 = c_1' \tag{34}$$

$$s_1' \in \pm\{0,1\}^{\varepsilon(\gamma_2+k)+1}, \qquad s_2' \in \pm\{0,1\}^{\varepsilon(\lambda_2+k)+1} \tag{35}$$

$$s_3' \in \pm\{0,1\}^{\varepsilon(\gamma_1+l_p+k+1)+1}, \qquad s_4' \in \pm\{0,1\}^{\varepsilon(2l_p+k)+1} \tag{36}$$

3.4 The Deposit Protocol

The deposit protocol involves the merchant and the bank B_j as follows:

a. The merchant sends to the bank B_j the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ on the message m' and the group blind signature $\sigma = (c, s_1, s_2, s_3, s_4, A, B, D)$ of the message m.

b. The bank B_j first verifies the validity of the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ using the same operations as the merchant (see step 5 from subsection 3.3).

c. If the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ is valid, the bank B_j verifies the validity of the group blind signature $\sigma = (c, s_1, s_2, s_3, s_4, A, B, D)$ using the same operations as the merchant (see step 2 from subsection 3.3).

If the group blind signature σ is valid and the coin m was not deposited before, the bank B_j accepts the coin m and then the merchant sends the goods to the customer.

If the coin m was deposited before, double spending is found. Then the bank B_i and the trusted third party can identify the identity of the dishonest customer.

3.5 The Trace Protocol

The bank B_j can legally trace the customer of a paid coin with the help of the Central Bank. The bank B_j sends the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ and the group blind signature $\sigma = (c, s_1, s_2, s_3, s_4, A, B, D)$ of the message m to the Central Bank. The Central Bank will find the bank B_i who issued the coin m using the open protocol of the group blind signature scheme [13]. When the bank B_i is found, the bank B_i and the trusted third party will find the dishonest customer using the open protocol of group signature scheme [2].

To open a group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ and reveal the identity of the dishonest customer (e.g., double spender) who created a given group signature, the trusted third party performs the following steps:

a. Verifies the validity of the group signature $(c_1, s_1', s_2', s_3', s_4', T_1, T_2, T_3)$ with public key P' using the same operations as the merchant (see step 5 from subsection 3.3).

b. Computes $A_i' = T_1 / T_2^{x'} \bmod n$ and generates a proof that:

$$\log_{g'} y' = \log_{T_2} T_1 / A_i'.$$

c. Search through the group member list to get the identity of the customer C_i corresponding to A_i'.

4 Security Considerations

In this section we discuss some aspects of security of our off-line electronic cash system with multiple banks. We will state the theorems and sketch the proofs, showing that the proposed system satisfies the following properties: unforgeability of coins, security against money laundering and anonymity of honest customer.

Theorem 1. *If the group blind signature scheme is secure against forgery and the hash function H is collision-resistant, the e-cash system is secure against forgery of the coin.*

Proof. Since the group blind signature scheme is secure against forgery, this allows only the legal bank to generate the blind signature for the coin m. As the hash function H is collision-resistant, the customer cannot forge the coin m. Furthermore, from the property of coalition-resistance of a group blind signature scheme, the banks will not collude each other, such that the issued group blind signature could not be open by the Central Bank. □

Theorem 2. *Assuming that the group signature scheme and the group blind signature scheme are computationally secure, the off-line e-cash system with multiple banks is secure against money laundering.*

Proof. Since the trusted third party knows the relation between customer's identification and his secret key, money laundering is prevented. When money laundering happens, the trusted third party reveals the identity of dishonest customer using the trace protocol. □

Theorem 3. *The e-cash system achieves anonymity with respect to the bank, that is, it is infeasible for the bank to trace legal customers without the help of the trusted third party.*

Proof. Assuming that the group signature scheme and the group blind signature scheme are computationally secure, our system is secure against tracing a honest customer by the bank. Identifying the actual honest customer is computationally hard for everyone, but the trusted third party, due to the group signature scheme. Also, since the group blind signature σ can not give any information for the coin m, the bank can not link the blind coin with the identity of the customer. Therefore, it is infeasible for the bank to trace honest customers without the help of the trusted third party. □

5 Conclusion

In this paper we proposed an off-line electronic cash system with multiple banks based on a secure coalition-resistant group blind signature scheme. Our scheme is an extension of the electronic cash scheme of Fangguo. Also, the main benefits of our off-line electronic cash system, compared to the scheme of Fangguo, relate to the underlying group signature scheme's improved efficiency and provable security.

References

[1] G. Ateniese, G. Tsudik, Some open issues and new directions in group signatures, *Proceedings of Financial Cryptography (FC'99)*, Anguilla, British West Indies, 1999, 196-211.

[2] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik, A Practical and Provably Secure Coalition-Resistant Group Signature Scheme, *Proceedings of Crypto 2000*, Santa Barbara, USA, 2000, 255-270.

[3] E. Brickell, P. Gemmel, and D. Kravitz, "Trustee-based tracing extensions to anonymous cash and the making of anonymous change", *Proceedings of The 6th ACM-SIAM*, pp. 457-466, 1995.

[4] J. Camenisch, M. Michels, A group signature scheme with improved efficiency, *Proceedings of Asiacrypt'98*, Beijing, China, 1998, 160-174.

[5] J. Camenisch, M. Stadler, Efficient group signature schemes for large groups, *Proceedings of Crypto'97*, Santa Barbara, USA, 1997, 410-424.

[6] S. Canard, and J. Traore, "On fair e-cash systems based on group signature schemes", *Proceedings of ACISP 2003*, pp. 237-248, 2003.

[7] D. Chaum, Blind signatures for untraceable payments, Proceedings of EUROCRYPT'82, pp. 199-203, 1983.

[8] H. Choi, F. Zhang, and K. Kim, Electronic cash system based on group signatures with revokable anonymity. *Proceedings of Workshop of Korea Information Security Institute*, pp. 29-34, 2003.

[9] Z. Fangguo, Z. Futai, W. Yumin, An off-line electronic cash systems with multiple banks, Proceedings of Information Security for Global Information Infrastructures, pp. 461-470, 2000.

[10] M. Gaud, and J. Traore, "On the anonymity of fair off-line e-cash systems", *Proceedings of Financial Cryptography*, pp. 34-50, 2003.

[11] A. Lysyanskaya, Z. Ramzan, Group blind signature: A scalable solution to electronic cash, *Proceedings of Financial Cryptography (FC'98)*, Anguilla, British West Indies, 1998, 184-197.

[12] G. Maitland, C. Boyd, Fair electronic cash based on a group signature scheme, *Proceedings of ICICS 2001*, Xian, China, 2001, 461-465.

[13] C. Popescu, A Secure and Efficient Group Blind Signature Scheme, *Studies in Informatics and Control Journal*, vol. 12(4), 269-276, 2003.

[14] C. Popescu, Svein J. Knapskog, An Off-line Payment System Based on a Group Blind Signature Scheme, Proceedings of the 3rd Conference on Security and Network Architectures (SAR'04), La Londe - Cote d'azur, France, pp. 101-110, 2004.

[15] W. Qiu, K. Chen, and D. Gu, A new off-line privacy protecting e-cash system with revokable anonymity. *Proceedings of ISC 2002*, pp. 177-190, 2002.

[16] M. Stadler, J.M. Piveteau, and J. Camenisch, "Fair blind signatures", *Proceedings of Eurocrypt'95*, pp. 209-219, 1995.

[17] J. Traore, Group signatures and their relevance to privacy-protecting off-line electronic cash systems, *Proceedings of Information Security and Privacy*, Wollongong, Australia, 1999, 228-243.

[18] B. Von Solms, D. Naccache, On blind signatures and perfect crimes, *Computers and Security, 11*(6), 1992, 581-583.

Constantin Popescu, Horea Oros
University of Oradea
Department of Mathematics and Computer Science
Address: 1, Universităţii St., 410087, Oradea, Romania
E-mail: {cpopescu,horos@uoradea.ro}

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 393-398

Parallel Video Processing using mpiJava & JMF

Niall Purcell, Sabin Tabirca, Daniel C. Doolan

Abstract: This article describes a solution for parallel video processing based on the mpiJava and Java Media Framework (JMF) technologies. The article begins with a brief introduction of the work which is followed by a review of the problem and similar articles. This paper describes the architecture of the parallel system and some important features of it. Experimental tests show that the execution time is greatly improved to that of the sequential computation.

1 Introduction

Video processing is an important field in the area of multimedia technology. Many of the current applications, such as Final Cut Pro (FCP) and Adobe Premier (AP), that run on single processor machines are very demanding both on the processor and the main memory. For example the rendering time of a few hundred video frames can take hours depending of the transformations to be applied. Therefore, there is a need to create video applications that address these issues and make the rendering time of the processed movie shorter. Developing multiprocessor applications can be a solution to the speedup the execution and to obtain reasonable rendering times. In this work a combination of technologies was used to create a parallel application that splits the input movie into small chunks to process. These include Sun's Java Media Framework(JMF) [1] for video processing and mpiJava which was developed as part of the HPJava project [3].

1.1 Problem Review

The applications FCP and AP provide a comprehensive set of tools for video processing. However, they can still be quite slow when rendering a video. Video frames maybe lost or distorted when the application is running on a slow machine. The need for distributed computing is obvious in this area. Several issues had to be overcome to create the application. One of the important issues that had to be overcome was the choice of languages used. The JMF and mpiJava libraries were chosen for this work, to create the parallel video processing application. JMF provides adequate features to be used in the area of multimedia. It contains MediaHandlers such as MediaPlayer, Player and Processor objects, that can be used to control and manipulate video [1]. These features are used to split the input movie into shorter movies and to extract each frame from the sub-movies for processing.

mpiJava is an object-oriented Java interface to the standard Message Passing Interface(MPI). It supports only the basic java datatypes such as byte, char and int. There is also a predefined datatype *MPI.OBJECT* for derived data types. It does not contain any means to supply graphical information like its C counterpart. The C implementation of MPI contains the Multi Processing Environment (MPE) libraries that can be used for performance analysis and as graphical visualization tools. This is one of the main drawbacks of the Java implementation of MPI [3]. Another requirement of the project is the ability to visualize the work in progress on individual processors. mpiJava was used to scatter the small movies onto processors and to gather the processed movies back to the root processor.

1.2 Related Work

Although parallel computing is a very efficient way to address video processing we have found only a handful of relevant publications that address this topic. In a very recent work Eidenberger [6] presents solutions based on LAN-connected PC workstations. The paper describes a prototype system implemented using free Java software and standard network protocols. Video data is streamed from and to processing hosts using IP multicast and Real-time transfer protocol based on on the Gigabit Ethernet. The system proves to be very flexible to scale it on several types of networks, however no experimental tests are given to show its efficiency.

Previous to this there had been some articles on Parallel Software-only Video Effects Processing systems(PSVP). The articles describe and discuss temporal and spatial parallelism. Mayer-Patel and Rowe [7] deal with spatial parallelism and the video effects system designed for internet video while in [8] they concentrate on temporal parallelism for the same system, parallelisation methods and control of data flow. Yang et.al. [9] developed a system based on pipelining computation for spatially and temporally parallelised video processing. It is based on

a scheduler that distributes the task at compile time on a network of workstation (NOW) to minimise the number of processors for a required task as well as to maximise the throughput.

2 Design

The parallel application is designed with software availability in mind and every attempt was made to use existing software. JMF and mpiJava are freely available downloads and provided adequate functionality to create the application.

2.1 mpiJava

mpiJava is a Java interface for the well known Message Passing Interface(MPI). It was created as part of the HPJava project [3]. It is implemented as a set of Java Native Interface(JNI) wrappers to native MPI packages. Java was seen as potentially a good language for parallel computing because it is simple, efficient and platform-neutral. mpiJava has several classes the most important of which is the Comm class. All communication functions in mpiJava are members of Comm or its subclasses. A communicator stands for a collective object shared by a group of processors. Processes communicate by addressing messages to their peers through the common communicator. Another important class is the Datatype class. This class describes the elements that can be passed as message buffers to the send, receive and other communication functions.

The program was designed to have several sections. Initially a sequential version was produced, which was then altered to provide a parallel version.

2.2 The GUI & Listener

The application consists of an interface created using Java Swing [2] components. It consists of several buttons, a slider, a time line, a list of applied filters and two areas to view a loaded video (see Figure 1). The slider and “add button” can be used to included different filters to the list and the time line provides a visual description of this list. Using these components we can create a sequence that contains initial and final times together with the associated filters.

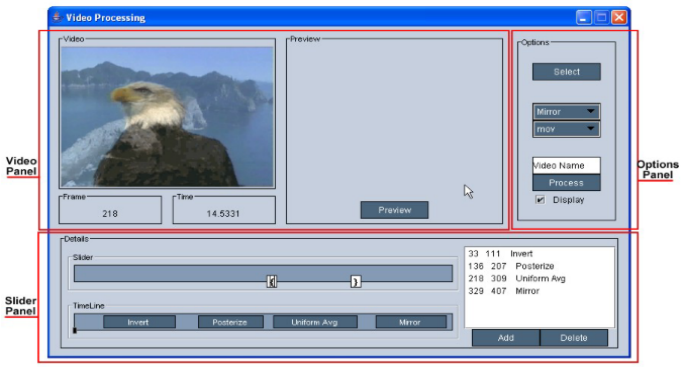


Figure 1: The GUI Elements

Initially the listener class associated with the buttons, tried to broadcast the action event among the processors. Although broadcasting the event object was successful, it wasn’t possible to re-trigger this distributed event object on processors other than the root. Therefore, the program continued to run only on the root processor.

To overcome this problem a work-around was found. Java provides a class Runtime and every application has a single instance of it. This allows the application to interface with the environment in which it is running [2]. The class also offers a method to execute a specified string command in a separate process. This method was used to start the parallel element of the application.

2.3 The Parallel Architecture

The architecture of the parallel section of the program is outlined in Figure 2. The root processor creates an array of realized MediaPlayer objects. There are several other arrays created that hold basic datatype data, including int arrays to hold the start and end frame numbers for each processor. These arrays, together with the array of MediaPlayers, are then scattered to the processors so that each processor receives a chunck of the video data.

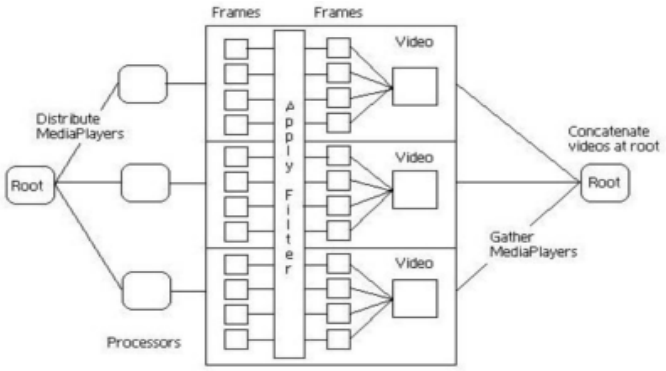


Figure 2: Architecture of system

It should be noted here that MediaPlayers are scattered as type MPI.OBJECT. Any java object that is either Serializable or Externalizable can be distributed using MPI.OBJECT as the specified datatype. The following code is an example of distributing MediaPlayer objects :

```
MPI.COMM_WORLD.Scatter(mediaplayers, rank, 1, MPI.OBJECT, mediaplayers, 0, 1, MPI.OBJECT, 0);
```

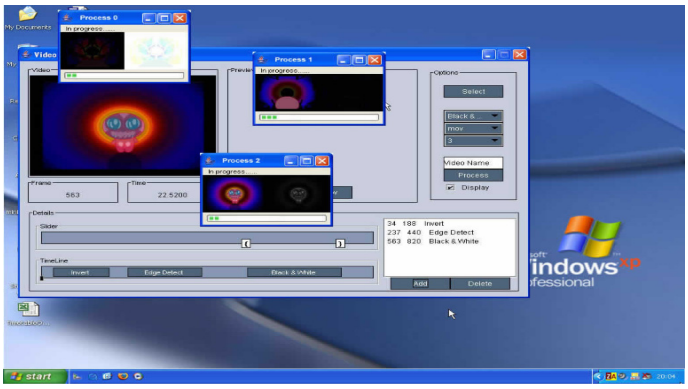


Figure 3: Computation on Each Processor

The processing involved is frame processing. A processor grabs each frame of video, one after another, from it's allocated section of video using JMF's FramePositioningControl class and the FrameGrabbingControl class. A filter can then be applied to a grabbed frame and the filtered image is then saved to disk in JPEG format. Sun Microsystems supplies code samples and solutions [4] with its JMF package. There is a java class available from this solutions page to create a movie file from a list of JPEG images. A modified version of this class was used to generate a movie file for each processor. By default the movie had to be in Quicktime format, however by adding a timestamp to each frame the movie could be saved in MSVIDEO format [5]. The class was modified to return a MediaPlayer object. Using the MPI.GATHER function, the MediaPlayer objects were returned to the root processor. The following is an example of gathering MediaPlayer objects :

```
MPI.COMM_WORLD.Gather(theMP, 0, 1,MPI.OBJECT,mediaplayers,rank,1,MPI.OBJECT, 0);
```

```
rank = MPI.COMM_WORLD.Rank(); size = MPI.COMM_WORLD.Size();
if(rank==0)preprocess(meadiaplayers, startFrames, endFrames);
MPI.COMM_WORLD.Scatter(mediaplayers, rank, 1, MPI.OBJECT, mediaplayers, 0, 1, MPI.OBJECT, 0);
MPI.COMM_WORLD.Scatter(startFrames, rank, 1, MPI.INT, startFrames, 0, 1, MPI.INT, 0);
MPI.COMM_WORLD.Scatter(endFrames, rank, 1, MPI.INT, endFrames, 0, 1, MPI.INT, 0);
MPI.COMM_WORLD.Bcast(rate, 0, 1, MPI.FLOAT, 0);
MPI.COMM_WORLD.Barrier();
filterMediaPlayer(mediaPlayer,starFrame, endFrame, filter);
MPI.COMM_WORLD.Gather(theMP, 0, 1, MPI.OBJECT, mediaplayers, rank, 1, MPI.OBJECT, 0);
MPI.COMM_WORLD.Barrier();
if(rank==0)(new MyConcat(mediaplayers, outFileName)).doIt();
MPI.Finalize();
```

Figure 4: Short Description of Parallel mpiJava Computation.

Application	Time (sec)
AP	193
FCP	205
Parallel Application (p=1)	254

Table 1: Premiere and CFP Vs. Project

An array of these MediaPlayer objects, one element for each processor, was then passed to a class that would concatenate the videos. This class was again available from the JMF samples and solutions page and was suitably modified for this project. The result was a video that had been processed using distributed computing (see Figure 4).

One of the nice features of the application is the ability to view the work taking place on each processor (see Figure 3). Each processor creates a JFrame that displays the original of the current frame and the current frame when filtered. When each processor completes filtering on the required number of frames, details of the processors are displayed. Details include the processing time, the number of frames processed and the machine name.

3 Experimental Tests

Testing took place under certain conditions to ascertain the performance benefits of using distributed computing for video processing. The application was tested against current video applications such as Final Cut Pro and Adobe Premiere. The sequential version of the program was also compared to the parallel version and the communication times of distributing MediaPlayer objects were recorded.

The first test completed involved the comparison between a professional application Adobe Premiere (AP), Final Cut Pro (FCP) and the parallel application run with one processor. As FCP and AP were unavailable on the cluster machine and subsequently the test had to be performed on a desktop machine (P4@2.66Ghz, 512Mb RAM, Windows XP). Table 1 illustrates the processing times of each application when the inversion transform is applied to each frame of the test video. The video used for this test had a resolution of 768x576 pixels, 906 frames @ 25 fps and a file size of 76.5Mbs. As can be seen from Table 1 the professional applications are faster than the parallel application run with one processor. However the difference between the speed of the applications is not hugely vast.

The second test involved running the program for different numbers of processors. The testing of the implementation was carried out on the Boole Centre Cluster. The cluster is made up of 50 Dell PowerEdge 1655MC modular servers consisting of 100 nodes. Each Blade server has dual PIII processors operating at 1.26Ghz, 1 gigabyte of system memory and a 18 gigabyte Ultra 160 SCSI HD. Each blade also has dual on-board gigabit Broadcomm Network Interface Cards (NICs). All the communication within the cluster is over gigabit networking. The video used for this test contained 4553 frames @ 15 frames per second(fps), had a resolution of 320x240 pixels and a file size of 76.9 Megabytes(Mb). Table 2 and Figure 5 illustrates the processing times in seconds of each processor. We can see that the application achieves a good load balancing with only marginal differences between the execution times. Moreover, the overall execution times reduce when the number of processors increases.

Other tests involved the analysis of communication times between processors. The communication times

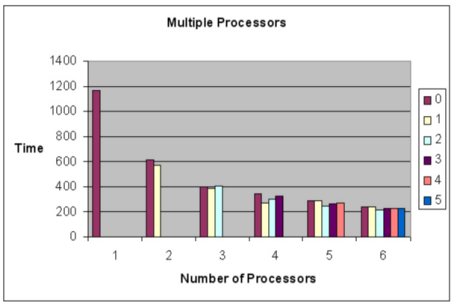


Figure 5: Graph of multiple processor test

Processors	Rank 0	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
1	1165.4					
2	610.14	570.47				
3	400.09	389.35	405.04			
4	347.35	276.27	297.85	323.89		
5	289.89	286.69	244.32	267.68	276.18	
6	236.83	237.05	216.41	223.83	225.89	228.21

Table 2: Processing times in seconds of each processor.

for scattering the MediaPlayer objects, start and end frame numbers for each section and the frame rate were recorded. The times for gathering the MediaPlayers were also recorded. Table 3 shows the difference between communication times for different file sizes. The test was completed using six processors. It can be concluded from Table 3 that the communication times are very small in comparison with the processing times. We can also see that the file size does not impact on performance when distributing MediaPlayer objects as the scatter and gather time values are very similar for different sizes of video.

File Size(Mb)	Comm	Rank 0	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
14	scatter	3.02	4.78	4.74	4.78	4.78	4.74
	gather	5.39	5.30	5.32	5.33	5.33	5.33
77	scatter	3.98	5.87	5.80	5.87	5.87	5.83
	gather	5.29	5.29	5.28	5.29	5.29	5.29
230	scatter	3.96	5.60	5.62	5.60	5.43	5.26
	gather	5.30	5.30	5.29	5.30	5.30	5.28

Table 3: Communication Times

The final experiments were carried out to find how the parallel application will run for very large video files. For that several files were used with the same video characteristics e.g. rate at 15 frames per second and resolution of 320x240 pixels. The execution times for p=1, p=2 and p=4 processors are presented in Table 4 for some file sizes of 77 Mb, 230 MB, 573 Mb and 1.12 Gb. Perhaps, the usefulness of this application can be seen on the execution times for the 1.12 Gb video. The rendering time came at 5 hours and 33 minutes when one single processor was used. In contrast carrying out the computation on four processors reduced the rendering time 1 hour and 38 minutes.

4 Conclusion

A parallel mpiJava application was develop to process video using JMF. Several interactive user interface elements were created to define how the video is being filtered. The system was also capable of allowing the root node to visualise the computation being carried out by each processor. Practical experiments have shown that this parallel application is a successful solution to reduce video render time.

Processors	**77 Mb**	**230 Mb**	**573 Mb**	**1.12 Gb**
$p = 1$	1165.4	3481.5	8692.4	19978.2
$p = 2$	610.14	1923.7	4397.2	11036.7
$p = 4$	347.35	1073.1	2591.7	5936.1

Table 4: Processing times in seconds for large video files.

References

[1] JMF Official Webpage, Sun Microsystems, `http://java.sun.com/products/java-media/jmf/`

[2] Java API, Sun Microsystems , J2SDK1.4.2_04 API, `http://java.sun.com/j2se/1.4.2/docs/api/`

[3] HP Java Project Webpage, Indiana University, 2003, `http://www.hpjava.org/mpiJava.html`

[4] JMF Samples, `http://java.sun.com/products/java-media/jmf/2.1.1/solutions/index.html`

[5] Code to convert to MSVIDEO format - `http://archives.java.sun.com/cgi-bin/wa?A2=ind0107&L=jmf-interest&P=R34660`

[6] Horst Eidenberger "Gigabit Ethernet-based Parallel Video Processing", *Proceedings of the 11th Internation Multimedia Modelling Conference (MMM 05*), IEEE Press, pp.358-363, 2005.

[7] K. Mayer-Patel, L.A. Rowe, "Exploiting Spatial Parallelism for Software-only Video Effects Processing", *Proceedings SPIE Multimedia Computing and Networking Conference*, SPIE Press, San Jose CA, pp. 28-39, 1998.

[8] K. Mayer-Patel, L.A. Rowe, "A Multicast Control Scheme for Parallel Software-only Video Effects Processing", *Proceedings ACM Multimedia Conference*, ACM Press, Orlando FL, pp. 409-418, 1999.

[9] M.T. Yang, R. Kasturi, A.A. Sivasubramaniam, "Pipeline-Based Approach for Scheduling Video Processing Algorithms on NOW", *IEEE Transactions on Parallel and Distributed Systems*, IEEE Press, vol. 14, no. 2, pp. 119-130, 2003.

Niall Purcell, Sabin Tabirca, Daniel C. Doolan
University College Cork
Department of Computer Science
Cork, Ireland
E-mail: {d.doolan,tabirca}@cs.ucc.ie

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 399-404

Multimedia Techniques for Watermarking Color Images

Monica Radulescu, Felicia Ionescu

Abstract: Digital color images are a kind of multimedia product that is considered to have a big impact on human perception and for this it is the most frequently used on Internet. Most of the watermarking techniques are dedicated to identification and copyright protection of this kind of multimedia material. The image representation through RGB color space components offers a lot of options to color image-watermarking algorithms. In this paper, two categories of watermarking techniques are proposed and compared. The first one extends the algorithms developed for grayscale images to color images by considering other color spaces that RGB, spaces obtained by linear transformations from RGB matrices. This kind of techniques treats a color image like three matrices corresponding to each color space component. The matrices are coded independently and the image is reconstructed. The second watermarking techniques category treats image color information in a compact way and in some cases it can be considered independent from pixels. Both categories have a common point in the strong connection between image quality and human visual perception. This paper presents existing algorithms and new ones, developed by the authors. We are looking for the perfect algorithm, which offers very high quality for the marked image and robustness to as many attacks as possible.

Keywords: watermarking, color space, human visual system, quality and robustness

1 Introduction

Because everyone can very easily access the Internet for obtaining information, the problem of copyright protection was taken into account and for this reason watermarking techniques were developed very quickly. This domain defines techniques for marking multimedia materials like digital images, video and audio sequences, text, 3D animation sequences, 3D models, etc.

The main requests of an image watermarking algorithm are keeping the host image quality at a high level and offering a mark robust to as many kinds of attacks as possible. Image quality is a subjective measure. It can be measured using a group of observers or approximated using perceptual measures like Watson measure and adaptive peak signal to noise ratio.

An attack is any kind of transformation which can be applied to an image but which doesn't destroy the image information. Attacks can be non-geometrical or geometrical because most part of the watermarking techniques doesn't resist to both kinds of attacks.

At the beginning, watermarking techniques were made for grayscale images. A pixel from a grayscale image has the value of its gray level intensity and the mark is inserted into that value.

Watermarking techniques use spatial or frequency domain (DCT, DFT, DWT, etc.), masking methods for finding the optimal gain factor and preserving image quality and distribution algorithms (we need to cover all image surface). Mark detection can be made using statistics or a marking image that will be extracted in the decoding process. (a logo or any interesting image)

Color images are represented in the RGB color space for display. This kind of images is a challenge for watermarking algorithms. Pixel information is three times bigger that the one from a grayscale image pixel. Obvious, we need to ask which one of those will be marked. All three of them or just one? First of all, using RGB space in watermarking algorithms doesn't offer the best results, because this space has very strong correlated components. Human visual system is less sensitive to blue channel distortion, so the mark can be added only to this component. This reduces color image watermarking to marking blue-scale images.

Watermarking color images techniques can be divided in two categories: methods that evolved from grayscale images algorithms, in which the components of the chosen color space are considered independent and dedicated algorithms for marking color images. In the last category of methods the whole image color information is used. The two categories don't exclude reciprocally. In this paper are presented watermarking techniques from both categories. The experimental results are analyzed and compared, searching the optimal color image watermarking algorithm.

2 Watermarking algorithms for color images extended from grayscale watermarking algorithms

At the beginning, watermarking techniques for color images used the components from RGB space. Because they didn't get very good results, they considered other color spaces which were obtained from RGB by linear transformations. (a perfect inverse transformation was needed for coming back in the RGB space). These color spaces are used for segmentation algorithms. Because JPEG standard uses YUV space and sub-sample the chrominance we can insert the mark into the luminance component. Most of the watermarking techniques for color images, obtained from classic methods, are using the luminance component. Other color spaces that can be used are YIQ, I1I2I3, XYZ (the linear part of the RGB - CIE transformation), CO (Color Opponency), YCM, etc.
In [1] an algorithm based on DCT is analyzed using many color spaces and in [2] the same analyze is done on a wavelet based algorithm. The only way to find the color component that has the best results is by experimental trying.
Watermarking techniques try to insert the mark into the host parameters that will be exposed to attacks as little as possible. Frequently used are the low and medium frequency coefficients (which are very little affected by lossy compression), feature points [3], very textured parts of an image, etc. The coefficients of a frequency band are obtained using a transformation like Fourier, Cosines (used in JPEG standard), wavelet (used in JPEG 2000 standard) Every algorithm has its own insertion mechanism. The mark can be inserted multiplicatively or additively, can be generated using a random generator or can be chosen for every image [4]. All these methods can be applied to grayscale and color images. The experimental results depend on the method we use, every one of these techniques trying to be very robust to attacks.
In this section is presented a color image watermarking technique based on wavelet transformation and its results after attacking the image using PhotoShop 7.0 This algorithm is similar to one in [2]. It is an adaptive algorithm based on wavelet and Hadamard transformations. Hadamard transformation is used to decorrelate the components of the low frequency band obtained after wavelet transform. Because we want an algorithm for color images we use the YUV space and we code separately the luminance and the chrominance. Every channel is coded with its own parameters but only the coding of the luminance channel is adaptive because to this channel the human visual system is more sensitive. The adaptability stays in recoding the host image several times with different parameters, until a good PSNR (adaptive measured) is achieved.
The experimental results obtained for this algorithm show an invisible mark and a very good quality marked image. The mark is also robust to many kinds of attacks like JPEG compression with a quality factor equal to 30, gif compression, wavelet compression (for JPEG 2000 standard), blur and sharpening filters, linear and non-linear filters, adding noise of different kinds, geometrical transformations reversed using PhotoShop tools, contrast, color or brightness adjustments, histogram equalization, thresholding, etc.
The main disadvantage for this kind of watermarking scheme is the powerful synchronization between the host image and the mark. If we broke this synchronization (exp. By applying the smallest geometrical attack) the mark can't be recover anymore. We also have to be careful to the marked image quality and have an adaptive gain factor. This algorithm has better results for grayscale images than for color images. The coding/decoding steps are complex and the proper color space and components have to be chosen by experimental results.

3 Color-based watermarking algorithms

Another kind of watermarking algorithms for color images is the category of color-based techniques. The mark is inserted into the color information of an image, respecting the same demands on quality and robustness as the classic algorithms. In this paper two methods from this category are proposed.
The first one is presented in [6] and is used more for the images that are going to be print and scan. For this reason the YCM space is used. The values of color components of a pixel are modified to obtain a maxim distortion of the luminance component. The mark is introduced additively. We tried for this technique two methods of detection. We used a correlation function for the coded components vector and the corresponding mark and we estimated the mean value of the image luminance and compared it with the one for original image. Each of these methods has better results for some categories of attacks. For example, if we rotate the marked image with four degrades the mark is correctly detected only with the first method of detection.
This watermarking technique has the disadvantage of geometrical synchronization between the host image and the mark. Besides, it isn't a blind technique because it uses for detection the original image.

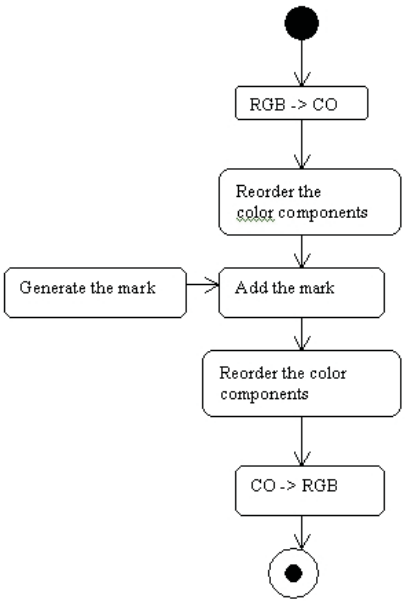


Figure 1: The activity diagram for the coding step Figure 2: Marked image using the presented method

The second method we propose organizes the color information after color components values (R, G, B) or after the ones of the CO space building an information vector that will be coded. In this way, the mark isn't geometrical synchronized with the image anymore and we achieve robustness to this kind of attack. The algorithm uses a random generated mark, which is additively introduced in the information vector. The activity diagram of the coding steps is given in Figure 1. The CO (Color Opponency) color space is obtained from RGB space through the next linear transformation:

$$A = R + G + B$$
$$BY = 2B - R - G$$
$$RG = R - 2G + B$$

The studies that were made proved that this color space is one of the best choices to hide small perturbation in colors because it is very close to the chromatic channels of the human visual system. The CO color space is used in texture or color recognition algorithms (exp. for human skin recognition). It has been also noticed that the distortion on BY direction are less observed by the human eye that the others [5] For this reason in the following watermarking algorithm the mark is inserted in RG and BY direction and for the last one the gain factor is three times bigger.

The experimental results obtained for the image from Figure 2 are the correlation function values for both of the marked direction for the marked image and for the original one. This relation gives the correlation function that was used:

$$(1)\quad \rho = \frac{\sum_{k=0}^{N} BY(i,j) * W(k)}{N},$$

where N is the number of different values of component A, which means 255*3.
In the following table are given the correlation function values obtained after applying more kinds of attacks to the marked image and to original, unmarked image. The threshold that says if an image is marked or not can be noticed from the given table. The experimental determination of the detection threshold is one of the main disadvantages of detection based watermarking algorithms.

Kind of attack	Correlation function values for marked image		Correlation function values for original image	
	BY	RG	BY	RG
No attack	23	23	-6	13
Scaling 50	18	22	-6	13
Rotation 59 ž	20	22	-6	13
Blur	16	21	-6	14
Gauss blur	16	21	-6	14
Gaussian noise addition	10	17	-7	11
Uniform noise addition	15	20	-6	13
Sharpening	24	23	-5	12
JPEG Q = 40	14	20	-5	13
.gif	21	18	-4	8
Brightness and contrast adjustments	-3	0	-4	0
Color adjustments	-6	-5	-4	-6

From the obtained experimental results we can notice that the mark isn't resistant to contrast, color or brightness adjustments. We have to say that these results were obtained for obvious modification of these properties. There is a limit of adjustments that can be made without loosing the mark. For the rest of attacks we have very good results. The great advantage of this method is the robustness to geometric attacks. This kind of robustness is very hard to achieve. It takes complex and special techniques. Using this method with the RGB components we obtained similar results.

4 Is it possible to have a "perfect" algorithm?

Comparing the two kinds of watermarking methods we noticed that they complete each other. Generally, algorithms extended from grayscale watermarking methods are based on frequency transforms and color-based algorithms are spatial. Color based algorithm has robustness to geometric attacks and wavelet based algorithm is resistant to color, contrast and brightness adjustments. None of the presented watermarking techniques affects image quality.
In this condition it is normal to ask ourselves if we can make one algorithm from this two to achieve robustness to as many attacks as possible without loosing quality. If an image coded using the method from [2] and the color based method presented here has a good quality, invisible marks and we can detect both marks inserted in it, we can say that a dual marking without individual disadvantages is possible.
In Figure 3 we have the image from Figure 2, coded using color based method, and marked using wavelet algorithm from [2]. For this image, no attacks, the correlation function values are 19 for BY direction and 21 for RG direction. We can also detect without problems the second mark inserted.
A dual marking method that uses an algorithm based on a frequency transform and a spatial algorithm has the advantage of resistance to many kinds of attacks. But if the two techniques aren't very well chosen the resulting image quality can be affected. For the presented combination of techniques the results are very good.
Because of the use of wavelet and Hadamard the algorithm is robust to non-geometric attacks and the mark is detected even after strong color or texture modifications. Non-geometric attacks implemented in special watermarking testing programs like Stirmark or Checkmark don't remove the mark. The only big weakness of this algorithm is to geometric attacks. Resynchronization after a geometric transformation is a long and complex process.
A spatial, geometric unsynchronized algorithm is fragile to attack that distortion the colors but not to geometric attacks.
By double coding an image using the presented methods we can achieve a very robust algorithm, which can detect

the mark even after many of the existent attacks were applied, which doesn't need the original image for detection and which offers a very good quality for the marked image. From these points of view we can consider the obtained algorithm " perfect ".

Figure 3: The image from Figure 2 with a second mark inserted based on [2]

But we still have the disadvantages from the semi-experimental choosing of the detection threshold, from the dependency between the coding parameters and the host image, from the dependency between results and host image, etc.

5 Conclusions

In this paper multimedia techniques for watermarking color images were analyzed. After we compared different algorithms for color images we can say that they evolved to dedicated algorithms. They have very good results, are more robust to more kinds of attacks and have very high quality for the marked image.
The algorithm we propose is resistant to compression, linear and non-linear filtering, adding noise (Gaussian, uniform, salt-and-pepper), geometrical attacks. These attacks are considered the common attacks. Besides it is resistant to texture addition, special effects based on PhotoShop, other kind of image processing techniques and special designed attacks. Image quality is not affected by the double coding and both of the coding methods are blind.
Especially for color images is very important how we choose the color space and the components that will be used in the marking process and is important to use an adaptive marking method. It can be noticed that different watermarking methods don't exclude each other as long as they use different kind of information for hiding the mark and that the results for a double coding has all the individual advantages without having the disadvantages too.
The great diversity of images and of image processing methods make almost impossible a perfect watermarking algorithm. Still, after image purpose, after the compromise made between robustness and quality we can choose the optimal algorithm or an optimal combination of algorithms.

References

[1] M Radulescu, F Ionescu, C Stoica, V Stoica , "Adaptive watermarking - two ways to define it", 2005.

[2] SAM Gilani, I Kostopoulos, AN Skodras , "Localized image watermarking based on feature point of scale-space representation" 2003.

[3] JS Seo, CD Yov , "Color image-adaptive watermarking" 2003.

[4] R Liu, T Tan , ”Content-based watermarking model” 1999.

[5] S Battiato1, D Catalano1, G Gallo1, R Gennaro , ”Robust watermarking for images based on color manipulation ” 1999.

[6] Al Reed, B Hannigan , ”Adaptive Color Watermarking”

[7] HD Cheng, XH Jiang, Y sun, J Wang , ”Color Image Segmentation- advances and prospects” 2000.

[8] M Kutter, FAP Petitcolas , ”A fair benchmark for image watermarking schemes” 1999.

[9] M Rabbani , ”Overview of State-of-the-Art watermarking technologies”

[10] SAM Gilani, AN Skodras, ”Multiple channels watermarking of color images” 2002.

Monica Radulescu, Felicia Ionescu
University “Politehnica” Bucharest
Faculty of Electronics, Telecommunications and Information Technology
Address: Bucharest, 2 Barbu Lautaru St., sector 1
E-mail: monnapit@gmail.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 405-409

About Using the Dirichlet Boundary Conditions in Heat Transfer Equation Solved by Finite Element Method

Adrian Sorin Roşca, Doina Roşca

Abstract: In the variational approach for heat equation, the Newmann, convection and radiation B.C. are already built in this expression. The Dirichlet (imposed temperature) must be applied after the matrix formulation was derived, for the analyzed system. This paper presents a study about the performances (precision and execution speed) using different methods to input the Dirichlet BC. These estimations are made on program which uses triangular elements, to solve the heat equation.

Keywords: heat transfer, Dirichlet Boundary Conditions, finite element

1 Introduction

From the available formulation which can be applied to solve the heat transfer equation by finite element approach, the variational one, rel.(1), is the most general and frequently used by commercial FEA software.

$$\Pi = \frac{1}{2}\int_V \left(k_x \left(\frac{\partial T}{\partial x}\right)^2 + k_y \left(\frac{\partial T}{\partial y}\right)^2 + k_z \left(\frac{\partial T}{\partial z}\right)^2 \right) dV - \tag{1}$$

$$- \int_V T q_1 \, dV - \int_{S1} T q_s dS + \int_{S2} \alpha \left(\frac{T^2}{2} - T \cdot T_m \right) dS$$

where: first integral form implements the conduction transfer in solid body, the second implements the internal heat sources, the third form inputs the Newmann BC on S1 part of the system boundary and the fourth form implements radiation or convection BC on S2 on the boundary, where the environment temperature is Tm.

As we can see, Newmann, Cauchy, radiation and convection BC are built in the previous expression, so they will be fitted in equation at the element level, after the discretisation process of this form. For Dirichlet BC (known temperature in some segments of the boundary), is necessary a special treatment which can be applied only after the assembly of the global system.

2 Applying the Dirichlet BC in matrix form of the system

After the transformation process of the integral form (1), in a algebraic equivalent system, which suppose: discretisation of integral form, expressing in the parametric space, applying stationary condition we derive the equation at the element level which have the aspect (2):

$$[K]^E \cdot [T]^E = [Q]^E \tag{2}$$

where $[K]^E$ is the conductivity matrix, $[T]^E$ is the nodal temperature vector and $[Q]^E$ is the vector with thermal loads.

For triangular orthotrop finite elements the matrix [K] has the dimension 3x3 and the vectors [T] and [Q] are 3x1. In fact the matrix [K] and the vector [Q] are derived by adding more matrix which are the equivalent of the terms from the integral form (1), so they contain all the BC less Dirichlet [1] . A diagram with the structure of matrix formulation is in figure 1.
Once obtained the matrix form (2), at element level, each matrix and vector is expanded at greater matrix of type nxn and the vector at type n. After this process they are assembled in a global linear system (3), which have the nodal temperatures as unknowns grouped in vector [T].

$$[K] \cdot [T] = [Q] \tag{3}$$

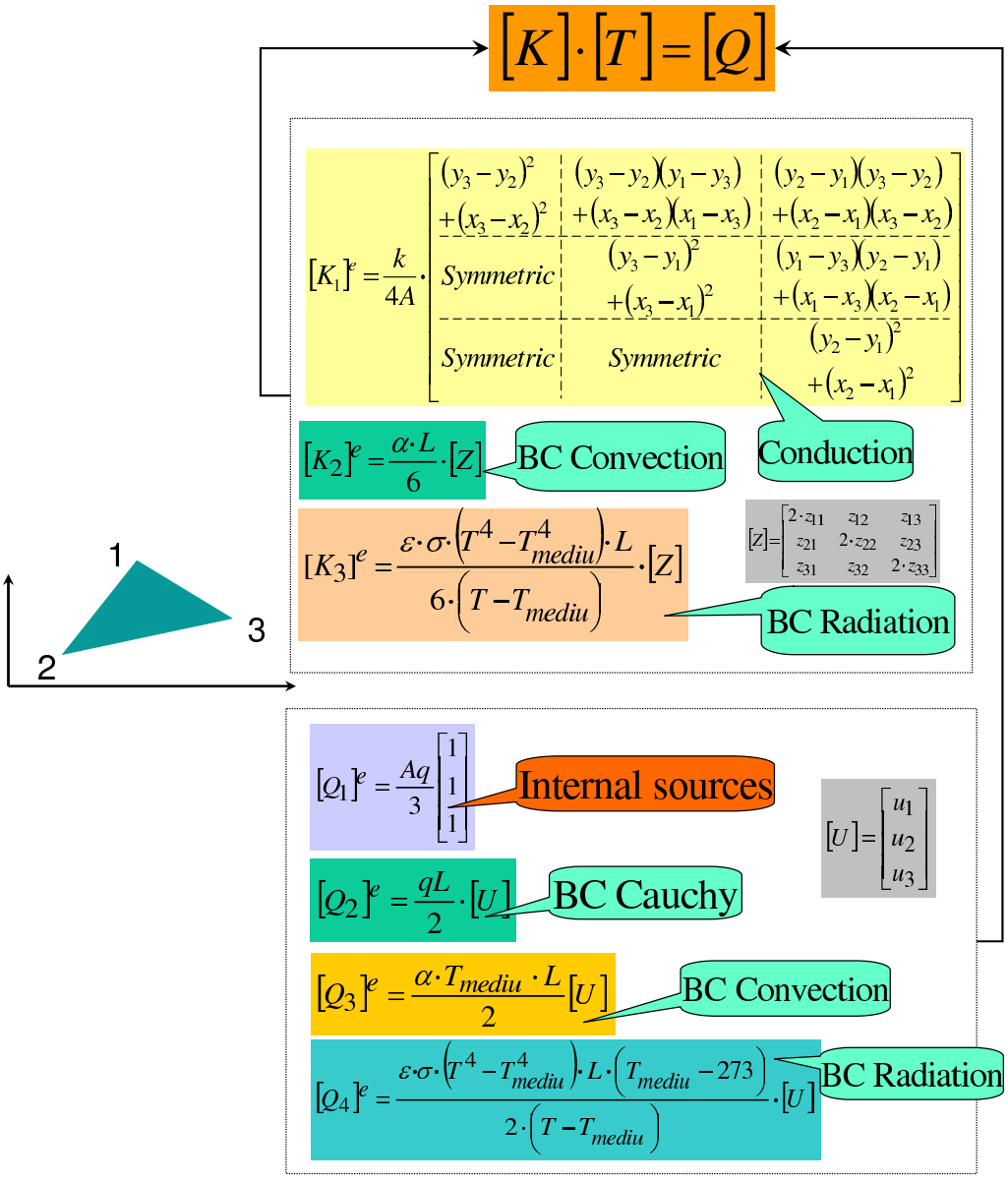


Figure 1: Matrix formulation

Considering only the conduction, the matrix [K] resulted after the assembly process is singular, so the solutions are trivial. To solve the system (3), we must introduce the BC. From the equation (1) we know for Dirichlet BC is necessary a special treatment, by one of the following methods:

- *Eliminating the line and column corresponding to affected node*

This sames to be the simplest way to introduce Dirichlet BC. Next, in (4), follows for a theoretic system with a single triangular element which have a imposed temperature in node 1, the matrix expression of the system:

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \tag{4}$$

$$\begin{aligned} k_{11}T_1 + k_{12}T_2 + k_{13}T_3 &= Q_1 \\ k_{21}T_1 + k_{22}T_2 + k_{23}T_3 &= Q_2 \\ k_{31}T_1 + k_{32}T_2 + k_{33}T_3 &= Q_3 \end{aligned} \tag{5}$$

In (5) are the developed equations of the system, where the unknowns are the nodal temperature T1 and T2, so the system can be write as (6):

$$\begin{aligned} k_{12}T_2 + k_{13}T_3 &= Q_1 - k_{11}T_1 \\ k_{22}T_2 + k_{23}T_3 &= Q_2 - k_{21}T_1 \\ k_{32}T_2 + k_{33}T_3 &= Q_3 - k_{31}T_1 \end{aligned} \tag{6}$$

To solve the previous system is much comfortable to eliminate the equation corresponding to node 1, as in equation (7):

$$\begin{bmatrix} k_{22} & k_{23} \\ k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} Q_2 - k_{21}T \\ Q_3 - k_{31}T_1 \end{bmatrix} \tag{7}$$

For large systems this method suppose complications in software code to rearrange the equations. This aspect become time consuming when are many nodes with Dirichlet BC which occupies arbitrary position in the system.

- *Cleaning the line and column corresponding to affected node*

This method uses a form similar to aspect (7) of the system at which is added an identity as a third equation so the system becomes (8).

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & k_{22} & k_{23} \\ 0 & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ Q_2 - k_{21}T_1 \\ Q_3 - k_{31}T_1 \end{bmatrix} \tag{8}$$

So in fact the elements from the line and column corresponding to a node with Dirichlet BC are zeroed and the element from the main diagonal is set to 1. Also the elements from the load vector are modified as be-fore. This method keeps the original positions of the nodes in the global matrix and load vector eliminating the rearranging process.

- *Dominant term on diagonal*

This method, from coding point of view is the simplest implying the following modifications in the global matrix system:
$\rightarrow$ at the term from main diagonal is added a number α with a very large value;
$\rightarrow$ the term from load vector is also multiplied by the same value α. Even simple this method can introduce some errors for the values of temperature which decrease when α grows.

3 Comparative study of the methods for Dirichlet BC

This study was based on a FEM application which solves heat transfer equation on 2D domain using triangular ortotrope elements. The application runs on Autocad 2002 as an external application and is compiled with ARX library. The final system can be solved by Gauss method using two variants: direct (faster but less precise) and total pivoting (slower but more precise).
The storing technique for global matrix uses dynamic allocation without sky line. For Dirichlet BC are available the cleaning line and column and dominant term method. Taking the advantage that the temperature values in any real system are less than $(2 \cdot 10^3 {}^\circ C$, the value used for α was (10^{100}), avoiding any situation when a product $\alpha \cdot T$ can be outside of double precision representation capabilities: $\approx -1.7 \cdot 10^{308} \ldots 1.7 \cdot 10^{308}$. For the test was chosen

[1] Due to limited space is not possible to develop them

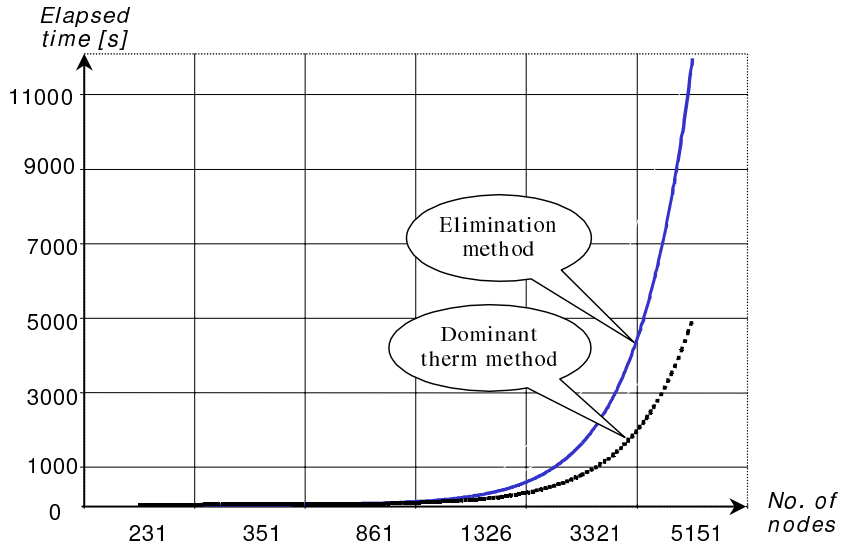


Figure 2: Speed comparison

a classic problem: a square plate having the side 2 units long, centred in origin, with Dirichlet BC $T = 0°C$ on the entire boundary. The plate have also an internal heat source distributed on the whole surface with the a unit rate. The material was considered isotrop with unit conductivity. For this problem Dhatt and Gouri in [4] indicates the analytic solution for the central point of the square as 0.2947. Six FEM models were tested with different number of nodes and elements. As we can see from figure 2, obvious influence on speed for the two methods used for Dirichlet BC are registered only for models with mode than 1,000 nodes.

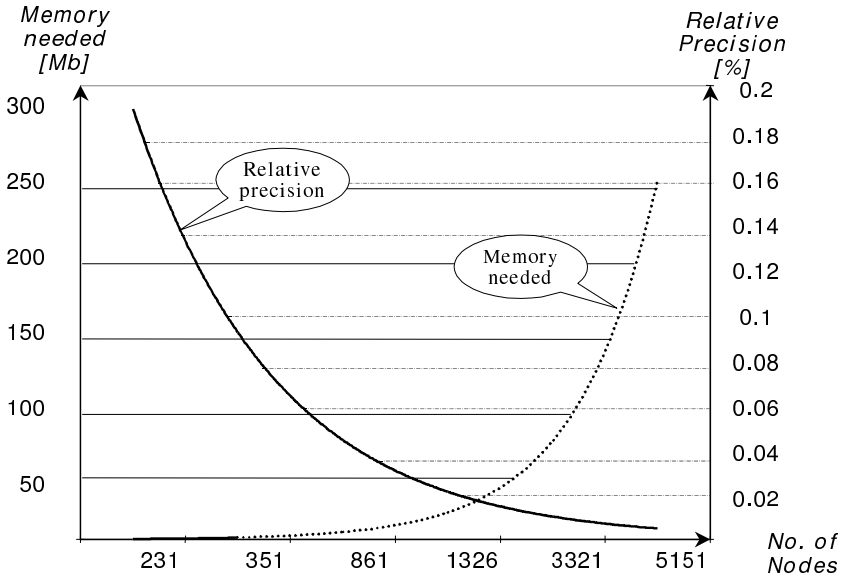


Figure 3: Precision and memory performances

At the model with 5151 nodes, the dominant diagonal term method reduces more than twice the necessary time to solve the problem. From the precision point of view both methods produces the same temperature values, so the chosen value represents a very good compromise: same precision but with appreciable speed advantages. Comparing with analytic solution, as we can se from fig 3, the best results were for the finest mesh with 5151 nodes (less than 0.02% differences), but with more than 250 Mb allocated for the problem.
Even at the roughest mesh with 231 nodes, the results were enough precise (0.2% differences) claiming an amount

of memory that can be allocated direct into the RAM of a normal PC. Analyzing these results a surprisingly conclusion can be extract: for the Dirichlet BC the dominant term method, offers for heat transfer problem, the best solution if the constant have the value around 10^{100}.

References

[1] Brauer, J.R., *What every engineer should know about finite elements analysis*, John Wiley & Sons, 1988.

[2] Bratianu, C., *Finite element modelling*, Icemenerg, 1985.

[3] Budovick, F.S., *Finite element mathematics with applications*, McGraw-Hill,1985.

[4] Dhatt, G., *The finite element method displayed*, John Wiley & Sons, 1984.

[5] Fagan, M., *Finite elements analysis, theory and practice*, McGraw-Hill, 1992.

[6] Liver, B., *Finite elements, an introduction for engineers*, Prentice-Hall, 1983.

[7] Steele, J.M., *Applied finite elements modeling, practical problem solving for engineers*, McGraw-Hill, 1989.

[8] Zienkiewicz, O., C., *The finite element method*, vol.1,2 ed.4, McGraw-Hill, 1989.

Adrian Sorin ROŞCA
University of Craiova
Faculty of Mechanics
Address: I.C. Bratianu str. 107, Craiova, Romania, 1100
E-mail: arosca@mecanica.ucv.ro

Doina ROŞCA
University of Craiova
Faculty of Horticulture
Address: A.I. Cuza str. 13, Craiova, Romania, 1100
E-mail: drosca@central.ucv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 410-416

Template for a Parallel - Distribute Application Based on a Messaging Service

Ernest Scheiber

Abstract: A Java template for parallel - distribute application based on a messaging service is presented. *TSpaces, JavaSpaces-Jini, Java Message Service* are between the messaging services that can be used. The logic of an application is the classical master - worker model. This template allows to translate some MPI or PVM applications into Java. It is developing a framework to support this template. An example is presented.

1 Introduction

In this note we present a template for parallel - distribute application based on a messaging service. The programming language is Java and a messaging service, as a network middleware enabling communications, is used. We have experimented with:

- *TSpaces 2.1.2* from IBM, [6];
- *Java Message Service 3.5 SP2*(JMS) from Sun, [4], [7];
- *Jini - JavaSpaces* from Sun, [12].

All these products are free of charges software, at least for a non commercial application.

TSpaces and *JavaSpaces* are Java implementations of the Linda computational model.

Java, a 10 years old programming language, is used in many projects, frameworks and products for parallel - distribute computing and for high performance computing. The Java technology is improving continuously. Java was designed to meet the real world requirement of creating interactive, networked programs. Java supports multithreaded programming, too.

A standard to develop parallel - distribute application are given by MPI [5], PVM [3] and BSP [3]. In this direction, in the Java world, there are functionally *mpiJava* [2], *MPJ Express* [7] – based on *mpiJava*, *JCluster* [1]. There are many other references but we don't have tested them.

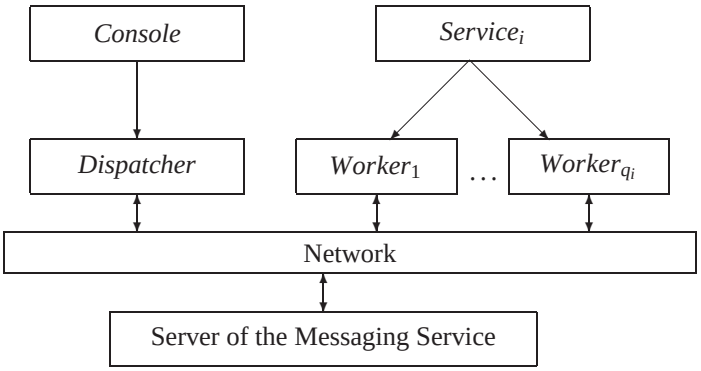
Other Java approaches for parallel - distributed programming are *ProActive* [13], *JavaParty* [8], *DPPEJ* [14], agent based frameworks like *JADE* [2].

There are some similarities between our approach with that reported for *Optimal Grid*, [16], and for a Jini based model, [15].

2 The template general description

We suppose that p computers in a network will perform the required computation. The application is composed from a dispatcher thread and q worker threads. It is the classical model of *master - worker.* On the computer $i \in \{1, \ldots, p\}$ there will be executed $q_i \geq 1$ worker threads ($\sum_{i=1}^{p} q_i = q$).

Between the dispatcher thread and the worker threads there are asynchronous message changes. These messages are kept by a server (TSpaces server, JMS provider, Jini JavaSpace service) until they are consumed.



The constructors of the dispatcher and of the worker classes are responsible to establish the connection with the server. The `run()` methods of the Java threads contain the specific activities to solve the given problem.
A message consumer (dispatcher or worker) waits until all the required messages are available - a barrier that solves the synchronization problems.
The data to be changed between the dispatcher and the workers are wrapped into objects. To distinguish between different kinds of data, a tag field may be introduced.
On each computer a Service program launches q_i worker threads, while on a computer a Console program starts the dispatcher thread.

3 A framework for the template

To support the above template, a framework is developing. The functionality of the framework is based on a *TSpaces* server. The role of the server is to keep the references of the involved computers. There is not a connection between this server and the messaging service server of an application.
A developer can deploy an application and launch it in execution. In order to deploy, an application must be arhivated with `jar`. A special attention must be done to the jar resources used by an application. In this case, these jar files will be put into a `lib` subfolder and the MANIFEST.MF file will contain the attribute `Class-path` with the list of jar files names.

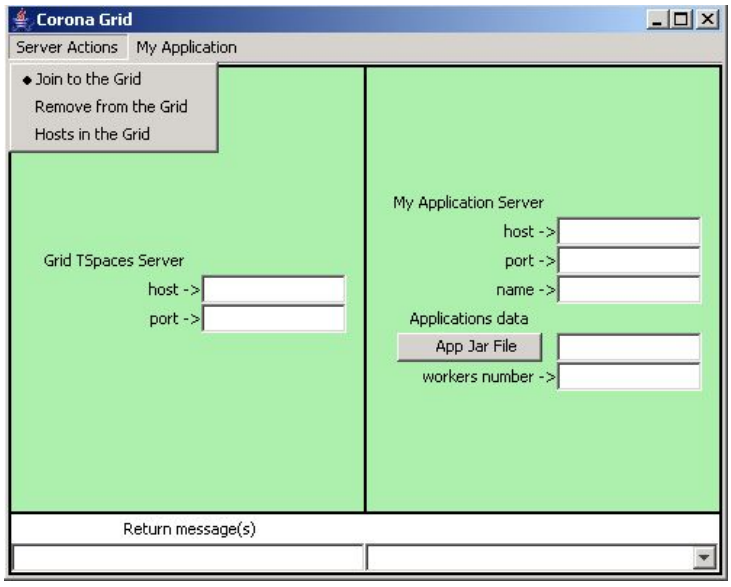


Figure 1: The window of the framework.

The working window of the framework (Fig. 1) contains two panels. The left panel is useful for joining to the

computing environment supported by the *TSpaces* server of the framework, while the right panel is useful to declare an application.
The messaging service server of an application must be started by the developer before deploying that application. The security is assured using a SecurityManager object with a proper policy file.

4 An example: The *Hello World* application

The dispatcher sends a greeting message to the workers. After receiving the message, a worker responds to the dispatcher indicating the name of the host computer. The results are written in text files.
We present simultaneously the TSpaces and the JMS versions of the Hello World application. It may be observed that, between the two versions, the codes relative to the communications differ only.
The dispatcher classes:

The TSpaces version

```
package tshello;
import com.ibm.tspaces.*;
import java.io.*;

public class Dispatcher extends Thread{
  private int tasks;
  private String appName;
  private PrintStream f;
  private TupleSpace ts=null;


  public Dispatcher(String appName,
      String host,int port,int tasks) {
    this.tasks=tasks;
    this.appName=appName;
    ts=startTupleSpace(appName,host,port);

    try{
      f=new PrintStream("dispatcher.txt");
    }
    catch(Exception e){
      System.out.println("File error :"+
        e.getMessage());
    }
  }

  public void run(){
    try{
                    // The dispatcher sends the greeting messages to the workers
      scatter();
                          // The dispatcher receives the response messages
      gather();
      ts.cleanup();
    }
    catch(Exception e){
      System.out.println(e.getMessage());
    }
  }

                    // The connection method to the application messaging server
  private TupleSpace startTupleSpace(
      String tsName,String host,int port){
    try{
      TupleSpace ts=null;
      Tuple active=TupleSpace.status(host,port);
      if((active==null)||
        (active.getField(0).getValue().
          equals("NotRunninng"))){
        System.out.println(
          "TSpace Server not available");
        System.exit(1);
      }
      ts=new TupleSpace(tsName,host,port);
    }
    catch(TupleSpaceException e){
      System.out.println("ConnectionException "+
        e.getMessage());
      System.exit(1);
    }
    return ts;
  }
```

The JMS version

```
package jmshello;
import javax.jms.*;
import java.io.*;

public class Dispatcher extends Thread{
  private int tasks;
  private String appName;
  private PrintStream f;
  private TopicConnection conn=null;
  private String jmsClient;

  public Dispatcher(String appName,
      String host,int port,int tasks) {
    this.tasks=tasks;
    this.appName=appName;
    conn=startConnection(host,port);
    jmsClient="Dispatcher"+appName;
    try{
      f=new PrintStream("dispatcher.txt");
    }
    catch(Exception e){
      System.out.println("File error :"+
        e.getMessage());
    }
  }

  public void run(){
    try{
      scatter();
      gather();
      conn.close();
    }
    catch(Exception e){
      System.out.println(e.getMessage());
    }
  }

  private TopicConnection startConnection(
      String host,int port){
    try{
      com.sun.messaging.TopicConnectionFactory
        cf=new com.sun.messaging.
          TopicConnectionFactory();
      cf.setProperty("imqBrokerHostName",host);
      cf.setProperty("imqBrokerHostPort",
        (new Integer(port)).toString());


      conn=cf.createTopicConnection();
    }
    catch(Exception e){
      System.out.println("ConnectionException "+
        e.getMessage());
      System.exit(1);
    }
    return conn;
  }
```

The TSpaces version

```
  private void scatter(){
    String mesOut="Hello from the dispatcher !";
    DataWrapper d=new DataWrapper(mesOut,0);


    try{
      Tuple multi=new Tuple();


      for(int i=0;i<tasks;i++){
        String dest=tsName+i;
        Tuple nextTuple=new Tuple(dest,d);
        multi.add(new Field(nextTuple));




      }
      TupleID[] ids=ts.multiWrite(multi);
    }
    catch(TupleSpaceException e){
      System.out.println("ScatterException : "+
        e.getMessage());
    }
  }

  private void gather(){
    DataWrapper d=null;
    Tuple tuple,template;
    try{









      for(int i=0;i<tasks;i++){
        String source=tsName+" "+i;
        template=new Tuple(source,
          new Field(DataWrapper.class));
        tuple=(Tuple)ts.waitToTake(template);
        d=(DataWrapper)tuple.getField(1).
          getValue();
        if(d.tag==1){
          f.println(d.mesaj);
        }

      }
      f.close();
    }
    catch(Exception e){
      System.out.println("GatherException : "+
        e.getMessage());
    }
  }
}
```

The JMS version

```
  private void scatter(){
    String mesOut="Hello from the dispatcher !";
    DataWrapper d=new DataWrapper(mesOut,0);
    Topic t=null;
    TopicPublisher publisher=null;
    try{
      TopicSession  session=
        conn.createTopicSession(false,
          Session.AUTO_ACKNOWLEDGE);
      for(int i=0;i<tasks;i++){
        t=new com.sun.messaging.Topic(appName+i);
        publisher=session.createPublisher(t);
        ObjectMessage
          m=session.createObjectMessage();
        m.setObject(d);
        publisher.publish(m);
      }

    }
    catch(Exception e){
      System.out.println("ScatterException : "+
        e.getMessage());
    }
  }

  private void gather(){
    DataWrapper d=null;
    Message msg=null;
    try{
      TopicSession session=conn.createTopicSession
        (false,Session.AUTO_ACKNOWLEDGE);
      Topic t=new com.sun.messaging.Topic
        ("Results"+appName);
      TopicSubscriber consumer=
        session.createDurableSubscriber(t,jmsClient);
      for(int i=0;i<tasks;i++){
        while((msg=consumer.receive())==null);
        if(msg instanceof ObjectMessage){

          ObjectMessage m=(ObjectMessage)msg;
          d=(DataWrapper)m.getObject();

          if(d.tag==1){
            f.println(d.mesaj);
          }
        }
      }
      f.close();
    }
    catch(Exception e){
      System.out.println("GatherException : "+
        e.getMessage());
    }
  }
}
```

The worker classes. The codes for the called methods `startTupleSpace` and `startConnection` are as in the corresponding Dispatcher classes and are omitted.
The codes are:

The TSpaces version

```
package tshello;
import com.ibm.tspaces.*;
import java.net.*;
import java.io.*;

public class Worker extends Thread{
  private int id;
  private TupleSpace ts=null;
  private String appName;
  private int tasks;
  private PrintStream f;


  public Worker(String appName,
      String host,int port,int tasks,int id){
    this.appName=appName;
```

The JMS version

```
package jmshello;
import javax.jms.*;
import java.net.*;
import java.io.*;

public class Worker extends Thread{
  private int id;
  private TopicConnection conn=null;
  private String appName;
  private int tasks;
  private PrintStream f;
  private String jmsClient;


  public Worker(String appName,
      String host,int port,int tasks,int id){
    this.appName=appName;
```

The TSpaces version	The JMS version

`this.id=id;`
`this.tasks=tasks;`
`ts=startTupleSpace(appName,host,port);`
 	
`try{`
`f=new PrintStream("worker"+id+".txt");`
`}`
`catch(Exception e){`
`System.out.println("File error :"+`
`e.getMessage());`
`}`
`}`
 	
`public void run(){`
`Tuple tuple=null;`
`DataWrapper d=null;`
`String source=tsName+id;`
`String dest=tsName+" "+id;`
`try{`
`// The worker receives the message from the dispatcher`
`Tuple template=new Tuple(source,`
`new Field(DataWrapper.class));`
`tuple=(Tuple)ts.waitToTake(template);`
 	
`d=(DataWrapper)tuple.getField(1).`
`getValue();`
`if(d.tag==0){`
`String msg="Mesage received by "+id+`
`" from the dispatcher:\n"+d.mesaj;`
`f.println(msg);`
`// . . . prepares and sends the response`
`InetAddress addr=`
`InetAddress.getLocalHost();`
`String s=addr.getHostName();`
`String mesOut="Hello from "+id+" at "+s;`
`d=new DataWrapper(mesOut,1);`
 	
`ts.write(dest,d);`
 	
`f.close();`
`}`
`}`
`catch(Exception e){`
`System.out.println("WorkerException: "+`
`e.getMessage());`
`}`
`}`
`}`
	`this.id=id;`
	`this.tasks=tasks;`
	`conn=startConnection(host,port);`
	`jmsClient=appName+id;`
	`try{`
	`f=new PrintStream("worker"+id+".txt");`
	`}`
	`catch(Exception e){`
	`System.out.println("File error :"+`
	`e.getMessage());`
	`}`
	`}`
	
	`public void run(){`
	`Message msg=null;`
	`DataWrapper d=null;`
	`TopicSession session=null;`
	`Topic t=null;`
	`try{`
	`session=conn.createTopicSession`
	`(false,Session.AUTO_ACKNOWLEDGE);`
	`t=new com.sun.messaging.Topic(appName+id);`
	
	`TopicSubscriber consumer=`
	`session.createDurableSubscriber(t,jmsClient);`
	`conn.start();`
	`while((msg=consumer.receive())==null);`
	`ObjectMessage m=(ObjectMessage)msg;`
	`d=(DataWrapper)m.getObject();`
	
	`if(d.tag==0){`
	`String msg="Mesage received by "+id+`
	`" from the dispatcher:\n"+d.mesaj;`
	`f.println(msg);`
	`InetAddress addr=`
	`InetAddress.getLocalHost();`
	`String s=addr.getHostName();`
	`String mesOut="Hello from "+id+" at "+s;`
	`d=new DataWrapper(mesOut,1);`
	`t=new com.sun.messaging.`
	`Topic("Results"+appName);`
	`m=session.createObjectMessage();`
	`m.setObject(d);`
	`TopicPublisher publisher=`
	`session.createPublisher(t);`
	`publisher.publish(m);`
	`conn.close();`
	`f.close();`
	`}`
	`}`
	`catch(Exception e){`
	`System.out.println("WorkerException: "+`
	`e.getMessage());`
	`}`
	`}`
	`}`

The launching classes. These programs are independent from the messaging server. The environment datas are fixed in property files with the content:

propConsole		propService	
Name		Name	
AppName		AppName	
Host		Host	
Port		Port	
TasksNumber	q	TasksNumber	q
		ComputerTaskNumber	q_i
		ComputerFirstTaskIndex	$\sum_{j=1}^{i-1} q_j,\ q_0 = 0$

The Console program	The Service program

`package ******;`
`import java.util.Properties;`
`import java.io.*;`
	`package ******;`
	`import java.util.Properties;`
	`import java.io.*;`

The Console program

```
public class Console{
  Properties p=null;

  public Console(String path2Prop) {
    try{
                                        // Loads the properties
      FileInputStream fis=new
        FileInputStream(path2Prop+"propConsole");
      p=new Properties();
      p.load(fis);
    }
    catch(Exception e){
      System.out.println(e.getMessage());
      System.exit(1);
    }
  }

  public static void main(String[] args){
    String fs=System.getProperties().
      getProperty("file.separator");
    String path2Prop;
                          // An argument may contain the path to the property file
    if(args.length>0)
      path2Prop=args[0]+fs;
    else
      path2Prop="";
    Console obj=new Console(path2Prop);
                                       // Set the properties
    String appName=obj.p.getProperty("AppName");
    String host=obj.p.getProperty("Host");
    String sPort=obj.p.getProperty("Port");
    String sTasks=obj.p.getProperty("TasksNumber");




    int port=Integer.parseInt(sPort);
    int tasks=Integer.parseInt(sTasks);


// The dispatcher is launched into execution

    Dispatcher dispacher=new Dispatcher(appName,
      host,port,tasks);
    dispacher.start();




  }
}
```

The Service program

```
public class Service{
  Properties p=null;

  public Service(String path2Prop)
    try{
      FileInputStream fis=new
         FileInputStream(path2Prop+"propConsole");
      p=new Properties();
      p.load(fis);
    }
    catch(Exception e){
      System.out.println(e.getMessage());
      System.exit(1);
    }
  }

  public static void main(String[] args){
    String fs=System.getProperties().
      getProperty("file.separator");
    String path2Prop;
    if(args.length>0)
      path2Prop=args[0]+fs;
    else
      path2Prop="";
    Service obj=new Service(path2Prop)
    String appName=obj.p.getProperty("AppName");
    String host=obj.p.getProperty("Host");
    String sPort=obj.p.getProperty("Port");
    String sTasks=obj.p.getProperty("TasksNumber");
    String sIndex=obj.p.getProperty(
      "ComputerFirstTaskIndex");
    String sComputerTasks=obj.p.getProperty(
      "ComputerTaskNumber");
    int port=Integer.parseInt(sPort);
    int tasks=Integer.parseInt(sTasks);
    int index=Integer.parseInt(sIndex);
    int computerTasks=Integer.parseInt(sComputerTasks);
// The required number of workers are
// launched into execution
    Worker[] worker=new Worker[computerTasks];
    for(int i=0;i<computerTasks;i++){
      worker[i]=new Worker(appName,host,
      port,tasks,index+i);
      worker[i].start();
    }
  }
}
```

The wrapper data class used by the above programs is

The Wrapper class

```
package ******;
import java.io.Serializable;

public class DataWrapper implements Serializable{
  int tag;
  String mesaj;

  public DataWrapper(String mesaj, int tag) {
    this.mesaj=mesaj;
    this.tag=tag;
  }
}
```

If the framework is used, then the class files of the above classes must be arhivated and the package name must be declared as the aplication messaging service server name.

5 Summary and Conclusions

An application template is pointed out to develop parallel - distribute applications based on a messaging service. The logic model of an algorithm is close to MPI or PVM models and even to the supersteps of a BSP algorithm [3].

Only the communication parts are replaced with that required by the messaging service software. The incapsulation of these communication tasks will be considered in the future.
A framework to support this template is developing. The current version is available from http:// cs.unitbv.ro/site/pagpers/scheiber.

References

[1] Z. BAOYIN, "Jcluster A Java Parallel Environment," Distributed with the software, version 1.0.5, 2005.

[2] F. BELLIFEMINE, G. CAIRE, T. TRUCCO, G. RIMASSA, "JADE Programmer's Guide," Distributed with the software, version 3.3, 2005.

[3] R. BISSELING, *Parallel Scientific Computation. A structured approach using BSP and MPI*, Oxford Univ. Press, 2004.

[4] B. CARPENTER B., G. FOX, S. H. KO, S. LIM, "mpiJava 1.2: API Specification," Distributed with the software, version 1.2.5, 2003.

[5] A. GEIST, A. BEGUELIN, J. DONGARRA, W. JIANG, R. MANCHEK, V. SUNDERAN, *PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge, MA, 1994.

[6] H. Q. MAHMOUD, "Getting Started with Java Message Service (JMS)," http://developers.sun.com, 2004.

[7] A. SHAFI, B. CARPENTER, M. BAKER, "MPJ Express: An implementation of MPI in Java," Distributed with the software, version 0_23, 2005.

[8] M. SNIR, D. OTTO, S. HUSS-LEDERMAN, D. WALKER, J. DONGARRA, *MPI: The Complete Reference.* MIT Press, Cambridge, MA, 1996.

[9] * * * , "TSpaces-User's Guide & Programmer's Guide," Distributed with the software, Version 2.1.2, 2000.

[10] * * * , *The J2EE 1.4 Tutorial*, Sun Microsystems Inc, 2004.

[11] * * * , *JavaParty* -documentation distributed with the software, version 1.07g, 2003.

[12] * * * , "Jini Technology Starter Kit Getting Started & More with v2.1," Distributed with the software, version 2.1, 2005.

[13] * * * , "ProActive Installation & User Guide," Distributed with the software, version 2.2, 2005.

[14] * * * , "Distribute Paralel Programming Environment for Java - DPPEJ: ReadMe, Extra docs," Distributed with the software, available at http://www.alphaworks.ibm.com, version 0.5, 2004.

[15] * * * , "Build a Compute Grid with Jini Technology," White paper, JINI_ComputeGrid_WP_FINL.pdf, www.jini.org, 2004.

[16] * * * , "Optimal Grid," available at http://www.alphaworks.ibm.com, 2004.

Ernest Scheiber
Transilvania University of Braşov
Department of Computer Science
Address: 50, Iuliu Maniu St., Braşov, Romania
E-mail: scheiber@unitbv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 417-422

A Cooperative Evolutionary Algorithm for Classification

Cătălin Stoean, Ruxandra Stoean, Mike Preuss, Dan Dumitrescu

Abstract: An evolutionary algorithm based on cooperative coevolution is applied to a classification problem, the Pima Indian diabetes diagnosis problem. Previous cooperative coevolution algorithms were developed for function optimization [1], optimizing agents behaviour [2] or modelling the behaviour of a robot in an unknown environment [3]. The aim of this paper is to integrate the cooperative approach into a learning classifier system and use it for solving a real-world problem of classification. To the best of our knowledge, there have been no attempts on applying cooperative coevolution specifically to classification. For each category of the classification problem, a sub-population evolves specific rules using a classical genetic algorithm. Sub-populations evolve simultaneously but independently; cooperation between them takes place only when the fitness of an individual in computed. Obtained experimental results encourage further investigation.

Keywords: genetic algorithm, cooperative coevolution, classification, evolutionary rules, diabetes mellitus

1 Introduction

Evolutionary models based on cooperative coevolution have been recently developed and applied for the optimization of difficult multimodal functions [1] and agent behaviour [3] and they have proven to be very successful. In [1], the performance of the cooperative based algorithm proved to be higher than the one of a typical evolutionary algorithm, as the cooperative one did not remain blocked into local optima, but always found the global one.

When a cooperative coevolutionary algorithm is applied to a problem, the first step is to find a natural decomposition of the problem into subcomponents. Then, each sub-problem is assigned to a sub-population, such that the individuals in a certain sub-population represent the potential subcomponents of the greater solution. Each sub-population is evolved simultaneously, but independently from the others. Collaboration is achieved only at the level of fitness evaluation; when the fitness of an individual is computed, collaborators from each of the other sub-populations are selected in order to form a complete solution which is evaluated [4].

In present paper, a new learning classifier system for binary classification is proposed. Decomposition of the problem is conducted with respect to the two classes; consequently two sub-populations are considered: one *evolves* a rule for the one class and the other for the opposite class. Each individual represents one rule. In the end of the algorithm, the best individuals from each of the two sub-populations represent the final rules. The cooperative approach is motivated by recent work [6] which indicated that two rules —one for each outcome— are sufficient to achieve good classification results.

When an individual of one sub-population is evaluated, its fitness is computed in correspondence with one individual from the other sub-population. The former shall be more similar to the objects of the training set which have the same outcome, and, at the same time, as different as possible from the latter individual, which represents the rule for the other outcome. In a sense, both subpopulations each evolve substitutes for the training set objects with one designated outcome.

The paper is organized as follows: next section presents some basics regarding cooperative coevolution, section 3 contains the detailed description of the proposed algorithm for classification and sections 4 and 5 present the diabetes diagnosis problem and the experimental results. The paper closes with the conclusions and some ideas for future work.

2 Cooperative Coevolution. Basic Concepts

Individuals in nature evolve by means of adaptation to the environment, part of which consists of other living beings itself, in particular of different groups or species. From this viewpoint, evolution is actually coevolution. Coevolution can be competitive, cooperative or both. Similarly in the evolutionary computational area, a interest has recently grown towards the extension of the powerful evolutionary algorithms to coevolutionary architectures. They are interesting indeed because they bring along a new idea for the fitness evaluation of an individual, i.e. in

relation to the other individuals in the population. As in nature, two techniques have been proposed: competitive and cooperative models.

We will briefly discuss the concepts underlying the latter. The first step towards a cooperative coevolutionary algorithm for a given problem is to decompose the problem into subcomponents and assign each subcomponent to a sub-population. Each sub-population evolves separately but concurrently with the others. Sub-populations collaborate only at the level of fitness evaluation, since each of them represents only a subcomponent of the problem and therefore a potential solution for every component in turn cannot be assessed apart from those of the complementary components. Therefore, every individual of every sub-population is evaluated by selecting collaborators from every other sub-population; a complete solution to the problem at hand is thus reached and its performance is computed and returned as fitness value of the current individual.

The main question in this process is the choice of collaborators. There are consequently three attributes regarding this selection whose values have to be decided when building a cooperative coevolutionary algorithm [4].

Collaborator selection pressure is the degree to which highly fit individuals will be chosen to form the complete solution to the problem, i.e. pick the best individual according to its previous fitness score, pick a random individual or select individuals based on classic selection schemes from each of the other sub-populations. **Collaboration pool size** is the number of collaborators that will be selected from each sub-population. Since each of these collaborations will have their own fitness score, the **collaboration credit assignment** will decide the value for the fitness of the current individual. There are three methods for this assignment, i.e. *optimistic* - the fitness of the individual whose fitness is computed is the value of its best collaboration, *hedge* - the average value of its collaborations is returned as fitness score and *pessimistic* - the value of its worst collaboration is assigned to the current individuals.

3 Proposed Algorithm

A formal representation for the binary classification problem is considered: training data is denoted by $\{(x_i, y_i)\}_{i\in\{1,2,\ldots,m\}}$; $x_i \in \mathbf{R}^n$ represents the input vector and $y_i \in \{0, 1\}$ is the *class* (or *outcome*).

3.1 Representation of Individuals

For each of the two classes, a sub-population of individuals is considered. The individuals in each sub-population represent IF-THEN rules; a rule contains n genes for each attribute of the input vectors and a last one which represents the class (0 or 1, in the binary case). All individuals in one sub-population have the same outcome, so the last gene does not suffer any modification during evolution.

3.2 Fitness Function

The distance between an object from the training set $x_i = (x_{i1}, x_{i2}, ..., x_{in}, y_i)$ and an individual $c = (c_1, c_2, ..., c_n, y_c)$ does not depend on the outcome and is given in (1).

$$d(c, x_i) = \sum_{j=1}^{n} \frac{| c_j - x_{ij} |}{b_j - a_j} \tag{1}$$

where a_j and b_j represent the lower and upper bounds of the j-th attribute. As usually the values for the attributes belong to different intervals, the distance measure has to refer their bounds.

When computing the quality of an individual c, a *collaboration* between c and **only one** individual e from the other sub-population - the best one from the previous generation or one randomly taken - is envisaged. The goal of the fitness function is to minimize the distance between c and all objects x_i of the training set with equal outcome and, at the same time, maximize the distance between the same objects and e. Consequently, the criteria are aggregated into the maximization problem in (2).

$$eval(c) = \frac{\sum_{i=1}^{m} h_c(e, x_i)}{1 + \sum_{i=1}^{m} h_c(c, x_i)} \tag{2}$$

where h_c is defined as follows:

$h_c : \mathbf{R}^{n+1} \times \mathbf{R}^{n+1} \rightarrow \mathbf{R}_+$,

$$h_c(a,b) = \begin{cases} d(a,b), & class(c) = class(b), \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

3.3 Algorithm Description

An evolutionary algorithm that learns characteristics for each of the two classes from the training data is further on presented. The rules that result after termination –the two individuals selected from each of the two classes– are applied to the test data.

Algorithm 1 Proposed evolutionary algorithm

```
t = 0;
initialize both sub-populations P1(t) and P2(t);
randomly select an individual from each of the sub-populations (b1 and b2)
repeat
   P1(t + 1) = evolve(P1(t));
   update b1
   P2(t + 1) = evolve(P2(t));
   update b2
until stop condition
```

The method "evolve" contains one generation from a typical genetic algorithm. Selection and then variation operators are applied to the population; resulted population is returned. The method could be described as follows:

function evolve(P)
evaluate population P;
apply selection for P;
apply recombination to the selected population;
mutate obtained population;
return resulting population;
end function

If collaboration selection pressure envisages the best individuals in each sub-population to be chosen for collaboration, then these best ones, denoted above by b_1 and b_2, will be determined as follows. At first, when the two sub-populations are initialized, b_1 and b_2 are randomly selected. Then, at generation t, the fitness of every individual from the first sub-population is computed in relation with b_2, while that of the individuals of the complementary sub-population is calculated with respect to b_1. Two individuals, one from each sub-population, that obtain the highest value for the fitness evaluation will now replace the b_1 and b_2 found at generation $t-1$.

Selection and Variation Operators

Tournament selection is employed. Mutation with normal perturbation and intermediate crossover are used. Naturally, crossover takes place only between individuals within the same sub-population. Mutation does not apply to the last gene (the outcome).

Stop Condition

The stop condition may refer to a predefined number of generations or a previously set number of generations that may pass without any improvement. The final b_1 and b_2 in Algorithm 1 represent the two rules that are to be applied to the test set.

4 Diabetes Diagnosis Problem

The Pima-Indian Diabetes data set comes from the UCI repository of machine learning databases [5]. All objects in the data set represent females of at least 21 years age, of Pima Indian heritage, living near Phoenix, Arizona,

USA. For each object in the data set there are eight attributes (either discrete or continuous) containing personal data, e.g. age, number of pregnancies, and medical data, e.g. blood pressure, body mass index, result of glucose tolerance test etc. The outcome is binary, either 0 (negative) or 1 (positive). 34.9% of the cases in the data set are assigned diabetes positive. The total number of cases is 768. No replacement or deletion of these values was undertaken in present paper.
The data is split into training and test sets. The task for proposed algorithm is to evolve two rules based on objects in the training set (one rule per outcome); these rules are then applied to the test set and the accuracy is computed as the percent of the patients from the test set correctly classified by the algorithm.
As an evolved rule represents a vector with eight values that correspond to the eight attributes, when a new object (a similar vector with eight attributes) from the test set is to be classified, the distance between that object and each of the two rules is computed. The outcome of the object coincides with the one of the closest rule to it.

5 Experimental Results

The first 75% of the cases represent the training set and the last 25% compose the test set; test sample cross-validation is conducted. As stated in previous section, two ways of establishing collaboration between an individual from one sub-population and one from the complementary sub-population are considered. In each of the two cases, same parameters of the evolutionary algorithm were considered - they are outlined in Table 1.

Population size*	No. of generations	Mutation strength	Mutation prob.	Crossover prob.
100	1000	100	0.1	0.4

* population size refers to only one subpopulation.

Table 1. Parameters of proposed evolutionary algorithm

The value of the mutation strength for a gene *i* directly depends on the size of the interval of the *i*-th attribute; in order to determine the value of the mutation strength for a gene, the size of the interval is divided into steps, the number of which corresponds to the value written in Table 1.
Figure 1 illustrates the progress of the accuracy obtained for both training and test sets when the best individual is considered for collaboration when fitness is computed.

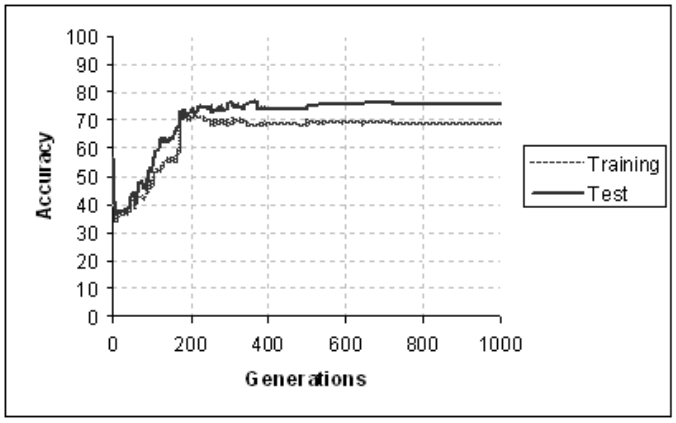


Figure 1: Accuracy obtained on training and test sets when collaboration is performed with the best individual in the complementary sub-population

Interestingly, the algorithm seems to work better when a random individual is selected for fitness evaluation (Figure 2). Support for the affirmation is not necessarily based on the weak start in Figure 1 - where it is probably just an unlucky initialization of the sub-populations - but on the accuracies reached during evolution and even on the final results (Table 2). The higher value of 77.08% on the test set is reached at generation 361 (Figure 2), while in the other case (Figure 1) the highest value reached in generation 364 is 76.56%. Another aspect that indicates the supremacy of the *random collaboration* choice refers to the number of fitness evaluations; in 30 runs, the average number of evaluations reached 879.992, while when collaboration was performed with the best one from

the complementary sub-population the average number of fitness evaluations again in 30 runs was 1.080.745. On the other hand, in both situations, the algorithm does not seem to need more than 400 generations to reach the optimum (see Figures 1 and 2), so if we set the number of generations parameter to 400 instead of 1000, the number of evaluations could be significantly decreased.
Searching for an explanation concerning the superior performance of *random collaboration*, we may conclude that for this problem, it is advantageous to let each sub-populations adapt to a set of individuals of the other sub-population instead of a probably rarely changing single best one.

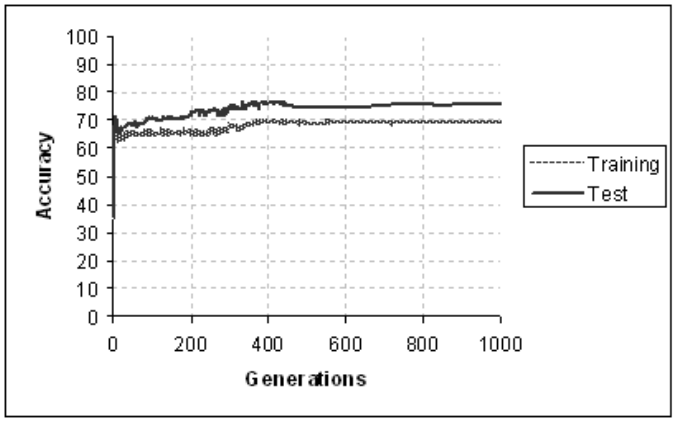


Figure 2: Accuracy obtained on training and test set when collaboration is performed with a random individual from the complementary sub-population

Pima Indian diabetes diagnosis task represents a largely used benchmark problem so, naturally, there are many results for comparison of the accuracy. Some of the best results found for the considered problem are outlined in Table 2.

Algorithm	**Repeats**	**Accuracy (%)**
Proposed algorithm & collaboration with best individual	30	**75.27**
Proposed algorithm & collaboration with a random individual	30	**75.4**
Best accuracy reached by proposed algorithm	1	**77.08**
EGGC algorithm in [6]	100	75.08
Neural Network (NN) in [7] with Prechelt's rules	30	65.5
Evolved NN in [8]	30	77.6

Table 2. Comparison to resulting accuracies of other methods for the diabetes diagnosis problem

Except the NN model with Prechelt's rules, all the others models used test sample cross-validation for the separation of training and test sets. Prechelt's rules regarding this separation imagine four ways of setting the training and test sets (in a percentage of 75% and 25%, respectively).

6 Conclusions and Future Work

In present paper, an evolutionary algorithm based on cooperative coevolution is integrated into a learning classifier system for binary classification and is applied for a real-world problem. The preliminary results indicate the high-quality of the proposed classifier.
Work in the near future envisages the generalization of the evolutionary classifier from binary to multi-class classification problems. At the same time, different strategies for collaboration between individuals from different sub-populations will be tested.
Another possibility to significantly improve proposed classifier regards the evolution of more than one rule for one class; this involves the use of a multimodal algorithm instead of a classical genetic algorithm for each sub-population.

References

[1] M. A. Potter and K. A. De Jong, ”A Cooperative Coevolutionary Approach to Function Optimization” *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, Springer, pp. 249-257, 1994.

[2] L. Panait, S. Luke, and R. P. Wiegand, “Biasing Coevolutionary Search for Optimal Multiagent Behaviors”, *IEEE Transactions on Evolutionary Computation*, to appear, 2006.

[3] M. A. Potter, L. A. Meeden and A. C. Schultz, ”Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergence of Specialists”, *Proceedings of The Seventeenth International Conference on Artificial Intelligence*, Morgan Kaufman, pp. 1337-1343, 2001.

[4] R. P. Wiegand, W. C. Liles, K. A. De Jong, "An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms", *Proceedings of GECCO 2001*, pp. 1235Ű1245, 2001.

[5] L. Prechelt, ”Proben 1 - a set of benchmark and benchmarking rules for neural network training algorithms”, *University of Karlsruhe, Institute for Program Structures and Data Organization (IPD)*, Tech. Rep. 21/94, 1994.

[6] C. Stoean, M. Preuss, R. Gorunescu, D. Dumitrescu, ”Elitist Generational Genetic Chromodynamics - a New Radii-Based Evolutionary Algorithm for Multimodal Optimization”, *Proceedings of the IEEE Congress on Evolutionary Computation*, UK, 2005, pp. 1839 - 1846.

[7] R. Smithies, S. Salhi, N. Queen, ”Adaptive Hybrid Learning for Neural Networks”, *Neural Computation*, vol. 16, no. 1, pp. 139-157, 2004.

[8] X. Yao, Y. Liu, ”A New Evolutionary System for Evolving Artificial Neural Networks”, *IEEE Transactions on Neural Networks 8(3)*, pp. 694-713, 1997.

Cătălin Stoean, Ruxandra Stoean
University of Craiova
Department of Computer Science
Address: 13, A. I. Cuza St., 200585, Craiova, Romania
E-mail: {ruxandra.stoean,catalin.stoean}@inf.ucv.ro

Mike Preuss
Dortmund University
Department of Computer Science
Address: 20, Joseph-von-Fraunhofer St., 44227,
Dortmund, Germany
E-mail: mike.preuss@cs.uni-dortmund.de

Dan Dumitrescu
Babes-Bolyai University, Cluj-Napoca
Department of Computer Science
Address: 1B, Mihail Kogalniceanu St., 400084,
Cluj-Napoca, Romania
E-mail: ddumitr@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 423-428

Evolutionary Multi-class Support Vector Machines for Classification

Ruxandra Stoean, Cătălin Stoean, Mike Preuss, Dan Dumitrescu

Abstract: Evolutionary support vector machines represent a new learning technique that we recently developed as a hybridization between support vector machines and evolutionary algorithms, regarding the discovery of the optimal decision function within the former. The new approach has proven to be successful as binary classification problems have been concerned. Present paper presents the extension of the aforementioned technique to the more frequent case of multi-class classification. Validation of evolutionary multi-class support vector machines is performed on the well-known benchmark problem of Fisher's Iris plants classification and results demonstrate the promise of the new approach.

Keywords: evolutionary support vector machines, multi-class classification, one-against-one method, Iris data set

1 Introduction

Support vector machines (SVMs) are a state-of-the-art learning technique which has proven its suitability in different real-world problems from fields like classification and regression [3], [9], [10]. Learning is regarded in a geometrical fashion and SVMs aim at building the decision hyperplane that separates between categories to be learnt. They were originally designed for binary learning problems and have been later extended to handle multi-class tasks, as well.

Although they represent a very successful and appreciated learning technique, the discovery of the separating hyperplane within is somewhat difficult to grasp and perform. This is why we decided to use the easily applied yet powerful evolutionary algorithms within SVMs in order to solve the optimization problem of determining the best separating surface. The new hybridized technique, called evolutionary support vector machines (ESVMs), has been applied to binary classification and has proven to be successful in the medical task of diabetes mellitus diagnosis [11], [12].

Present paper extends the new technique from binary to multi-class classification. From the wide range of SVM methods that have been built in this respect, the ONE-AGAINST-ONE method is chosen as a basis. The standard technique combines binary classifiers using a voting system. Consequently, the new technique combines evolutionary binary classifiers and the majority vote across the obtained classifiers will decide the class for a new point. The new approach is validated on Fisher's Iris data set.

The paper is structured as follows. Section two presents the concepts within binary SVMs for classification. Section three introduces binary ESVMs for classification. Section four explains the standard extension to multi-class SVMs. Section five presents the multi-class ESVMs; the experimental setup and results on the Iris data set are also depicted. Conclusions and ideas for future work are reached in the final section.

2 Binary support vector machines for classification

Suppose given training data is $\{(x_i, y_i)\}_{i=1,2,...,m}$, where every $x_i \in R^n$ represents an input vector (point) and each y_i its output (label).

If one supposes at first that the two subsets of input vectors labelled with $+1$ and -1, respectively, are linearly separable, then the positive and negative training vectors will be consequently separated by the hyperplane $\langle w, x\rangle - b = 0$, where $w \in R^n$ is the normal to the hyperplane, $b \in R$ and $\frac{|b|}{\|w\|}$ is the distance from the origin to the hyperplane. Hence, two data subsets are linearly separable iff there exist $w \in R^n$ and $b \in R$ such that:

$$\begin{cases} \langle w, x_i\rangle - b > 0, & y_i = 1, \\ \langle w, x_i\rangle - b < 0, & y_i = -1, i = 1, 2, ..., m. \end{cases} \tag{1}$$

Moreover, according to [1], two data subsets are linearly separable iff there exist $w \in R^n$ and $b \in R$ such that:

$$\begin{cases} \langle w, x_i\rangle - b > 1, & y_i = 1, \\ \langle w, x_i\rangle - b < -1, & y_i = -1, i = 1, 2, ..., m. \end{cases} \tag{2}$$

Following the *Structural risk minimization* principle [13], in order to generalize well, the support vector machine must provide a hyperplane that separates the training data with as few errors as possible and, at the same time, with a maximal margin of separation. One subsequently obtains the optimization problem (P_1):

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2}, \\ \text{subject to } y_i(\langle w, x_i \rangle - b) \geq 1, i = 1, 2, ..., m. \end{cases} \tag{3}$$

where $\frac{2}{\|w\|}$ is the value of the margin.
If one supposes now that the training data set is nonseparable, it is obviously not possible to build a separating hyperplane without any classification errors. However, construction of an optimal hyperplane that minimizes misclassification would be important under these conditions [7]. Previous ideas can be extended to handle this new situation as follows. Based on the fact that any training data point has a deviation from its supporting hyperplane, i.e. from the ideal condition of data separability, of $\frac{\pm \xi_i}{\|w\|}$, where $\xi_i > 0$, the positive variables are introduced in the separation condition, which becomes [4]:

$$y_i(\langle w, x_i \rangle - b) \geq 1 - \xi_i, i = 1, 2, ..., m \tag{4}$$

Therefore, for a training data point to be erroneously classified, its corresponding ξ_i must exceed unity. Simultaneously with (4), sum of misclassifications must be minimized. As a consequence, (P_1) changes to (P_2):

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{\|w\|^2}{2} + C \sum_{i=1}^{m} \xi_i, \\ \text{subject to } y_i(\langle w, x_i \rangle - b) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = 1, 2, ..., m. \end{cases} \tag{5}$$

where C is a variable that assigns penalties for errors.
The concepts can be extended even further to the construction of a nonlinear separating hyperplane for nonseparable data. Based on [5], training data can be nonlinearly mapped into a high enough dimensional space and linearly separated there. Suppose an input vector is mapped into some Euclidean space, H, through a mapping $\Phi : R^n \mapsto H$. It can be easily seen that within (P_2) all vectors in R^n appear only as part of dot products. The corresponding vectors in H should appear as part of dot products in the new optimization problem formulation, as well. Therefore, the equation of the separating hyperplane in H becomes $\langle \Phi(w), \Phi(x_i) \rangle - b = 0$, where $\Phi(w)$ is the normal to the hyperplane. And the squared norm, $\|w\|^2 = \langle w, w \rangle$, changes to $\langle \Phi(w), \Phi(w) \rangle$. The difficulty lies in how to pick the proper function Φ. If there were a kernel function K such that $K(x,y) = \langle \Phi(x), \Phi(y) \rangle$, where $x, y \in R^n$, one would use K in the training algorithm and would never need to explicitly even know what Φ is. At this point, the question is what kernel functions meet the condition above. The answer is given by Mercer's theorem from functional analysis [2]. The problem is that it may not be easy to check whether Mercer's condition is satisfied in every case of a new kernel. There are, however, a couple of classical kernels that had been demonstrated to meet Mercer's condition [2] and have been always used in practical applications, i.e. the polynomial classifier of degree p, $K(x,y) = \langle x, y \rangle^p$, and the radial basis function classifier, $K(x,y) = e^{\frac{\|x-y\|^2}{\sigma}}$. Consequently, the optimization problem (P_2) now changes to (P_3):

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \frac{K(w,w)}{2} + C \sum_{i=1}^{m} \xi_i, \\ \text{subject to } y_i(K(w, x_i) - b) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = 1, 2, ..., m. \end{cases} \tag{6}$$

Optimization problem (P_3) is next solved and once w and b are found, the class for a test vector x can be computed as $f(x) = sgn(K(w,x) - b)$.

3 Evolutionary binary support vector machines for classification

The evolutionary algorithm is embedded into the binary SVM at the level of solving the optimization problem (P_3), standardly approached by resorting to a generalized method of the Lagrange multipliers. Since the goal of the SVMs is to find the optimal parameters of the separating hyperplane such that it generalizes well, the corresponding target of the evolutionary algorithm is to evolve w and b for the same optimization task.

The evolutionary technique is a standard genetic algorithm [6] and its components are chosen as follows.

Chromosome representation
A chromosome comprises in its structure w, b and the ξ_is:

$$c = (w_1, ..., w_n, b, \xi_1, ..., \xi_m) \tag{7}$$

Proposed evolutionary algorithm thus includes the training errors in the structure of the chromosome.

Initial population
Chromosomes are randomly generated following a uniform distribution, such that $w_i \in [-1,1], i = 1,2,...,n$, $b \in [-1,1]$ and $\xi_j \in [0,1], j = 1,2,...,m$. When using a radial kernel, b can be directly generated in the $[1,2]$ interval in order to avoid unnecessary and expensive search in the solutions space until reaching the same region, as it cannot have a lower value than one, due to the expression of the kernel function.

Fitness evaluation
The expression of the fitness function is considered as follows:

$$f(c) = f(w_1, ..., w_n, b, \xi_1, ..., \xi_m) = K(w,w) + C\sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} [t(y_i(K(w,x_i) - b) - 1 + \xi_i)]^2, \tag{8}$$

where

$$t(a) = \begin{cases} a, & a < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

One is led to minimize(f(c), c).

Genetic operators
The operators were chosen experimentally. Tournament selection is used. Intermediate crossover and mutation with normal perturbation are considered. Mutation is restricted for errors, preventing the ξ_is from taking negative values (and b from taking values less than unity, if the second manner of initialization is adopted).

Stop condition
The algorithm stops after a predefined number of generations. In the end, it obtains the parameters of the hyperplane, i.e. w and b. Errors on training set may also result from the algorithm, i.e. those corresponding to $\xi_i > 1$, $i = 1,2,...,m$, if desired. ESVMs are therefore able to self determine their errors on the training set if proper values for parameters are chosen, but at the expense of loss of higher accuracy.

4 Multi-class support vector machines for classification

One classical and very successful method for multi-class classification within SVMs is called the ONE-AGAINST-ONE (1-a-1) technique [8]. Suppose the classification problem is k-class. Then, 1-a-1 considers $\frac{k(k-1)}{2}$ binary SVMs, where each machine is trained on data from every two classes, i and j, where i corresponds to *1* and j to *-1*. After the decision surfaces are obtained, the following voting method is used to determine the class for a test vector x. For every support vector machine, the class of x is computed as in binary support vector machines, following the sign of the decision function applied to that vector. Then, if the sign says x is in class i, the vote for the i-th class is incremented by one; conversely, the vote for class j is added by one. Finally, x is taken to belong to the class with the largest vote. In case two classes have identical number of votes, the one with the smaller index is selected.

5 Evolutionary multi-class support vector machines for classification. Modelling the Iris data set

The construction of 1-a-1 multi-class ESVMs is straightforward. $\frac{k(k-1)}{2}$ binary ESVMs are built for every two classes; voting is then conducted. Validation of the evolutionary approach to multi-class SVMs is achieved on the Iris data set from the UCI repository of machine learning databases.

5.1 Fisher's Irises

The Iris data set is a well-known benchmark problem for multi-class classification. It contains 3 classes of 50 instances each, where each class refers to a type of iris plant, namely Iris Setosa, Iris Versicolour and Iris Virginica. There are 4 attributes referring sepal length, sepal width, petal length and petal width. Visualization of the Iris data set can be of great help in getting a better picture of how objects are positioned and a deeper understanding of the results, i.e. the separating surfaces and related errors. Therefore, a plot of the data can be seen in Figure 1, courtesy of *http://www.ku.edu/cwis/units/IPPBR/java/iris/irisglyph.html*. Petal length is represented on the *x*-axis, sepal length on the *y*-axis and sepal width is on the *z*-axis. Petal width is represented by the angle of the glyph. Each of the three species of iris is represented by a different color.

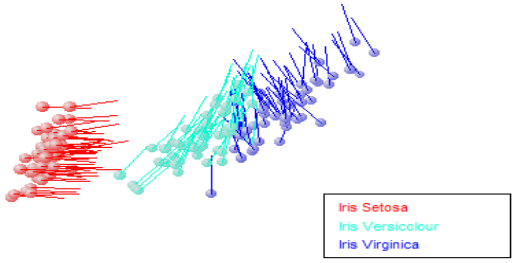


Figure 1: Visualization of the Iris data set as a glyph graph

5.2 Parameter setting

A radial kernel was chosen. Values of parameters for the three binary SVMs and for the corresponding evolutionary algorithms were chosen experimentally, with the same values for each classifier. They are given in Table 1. Values of parameters were chosen in favour of a high accuracy at the cost of the ability to self-determine the training errors. Different mutation parameters for variables of the hyperplane and errors have been appointed based on experimental expertise.

C	1
σ	1
Population size	100
Number of generations	1000
Crossover probability	0.3
Mutation probability	0.4
Mutation probability for errors	0.3
Mutation strength	0.03
Mutation strength for errors	0.05

Table 1. Values for parameters of the three binary ESVMs

5.3 Experimental results

The Iris data set was split into 70% training - 30% test. First, test sample cross-validation with equally distributed cases for each class both in training and test sets, taken in sequential order, was performed. Then random cross-validation was conducted. Normalization of data is also undertaken. The algorithm was run ten times for each type of cross-validation; obtained training accuracy for the three evolutionary binary classifiers in both cases are given in Tables 2 and 3. Results seem consistent with the visualization in Figure 1.

	Setosa vs Virginica	Setosa vs Versicolour	Versicolour vs Virginica
Training accuracy	100%	100%	93.27%
Standard deviation	0%	0%	1.17%

Table 2. Training accuracy for each of the evolutionary binary classifiers in 1-a-1 multi-class ESVMs with test sample cross-validation

	Setosa vs. Virginica	Setosa vs. Versicolour	Versicolour vs. Virginica
Training accuracy in mean	100%	100%	95.42%
Standard deviation	0%	0%	1.36%

Table 3. Training accuracy for each of the evolutionary binary classifiers in 1-a-1 multi-class ESVMs with random cross-validation

After the three classifiers were obtained, voting was conducted and accuracy on the test set was obtained as in Table 4.

	Test sample cross-validation ESVM	Random cross-validation ESVM
Test accuracy in mean	100%	95.77%
Standard Deviation	0%	1.26%

Table 4. Comparison of test accuracy for 1-a-1 multi-class ESVMs with test sample vs. random cross-validation

Comparison to results obtained by 1-a-1 standard multi-class SVMs on the Iris data set in [8] and [14] is given in Table 5. In the former, authors used ten-fold cross-validation and a shrinking technique, while in the latter random cross-validation was performed ten times with 90% training-10% test data split.

	Test sample 1-a-1 ESVM	Random 1-a-1 ESVM	10-fold 1-a-1 SVM	Random 1-a-1 SVM
Test Accuracy	**100%**	**95.77%**	97.33%	98.67%

Table 5. Test accuracy for the 1-a-1 multi-class ESVM compared to results by standard 1-a-1 methods

6 Conclusions and future work

Present paper extends the new technique of evolutionary support vector machines to multi-class classification. Hybridization is achieved with state-of-the-art ONE-AGAINST-ONE method within multi-class SVMs. Similarly to the binary case, there are many advantages of multi-class ESVMs over SVMs. ESVMs are definitely much easier. In the training stage, the evolutionary solving of the optimization problem enables obtaining w and b directly, while in the classical approach the equation of the optimal hyperplane is determined after Lagrange multipliers are found. Moreover, in the case of the classical technique, when using kernels in which Φ cannot be explicitly obtained, it is not possible to determine w and b at all - solely the class for a new vector can be predicted. The evolutionary method can also provide, if desired, which training data cannot be correctly classified, as errors are included in the structure of the chromosomes; the evolutionary support vector machines can self-determine their training error. Finally, accuracy on a benchmark real-world problem is comparable to those of state-of-the-art standard methods based on the same SVM approach to multi-class tasks. On the other hand, SVMs are faster, but future work envisages changes to the evolutionary algorithm in order to decrease computational time. For example, the inclusion of error gene deletion when they have already converged to zero will shorten the length of the chromosome and thus probably increase speed. ESVMs based on other state-of-the-art SVM approaches to multi-class classification are also envisaged in future work.

References

[1] R.A. Bosch, J.A. Smith, "Separating Hyperplanes and the Authorship of the Disputed Federalist Papers", *American Mathematical Monthly*, Volume 105, Number 7, pp. 601-608, 1998

[2] B. E. Boser, I. M. Guyon and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 11-152, Pittsburgh, PA, ACM Press, 1992

[3] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery* 2, 121-167, 1998

[4] C. Cortes, V. Vapnik, "Support Vector Networks", *Machine Learning*, 20:273-297, 1995

[5] T. M. Cover, "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition", *IEEE Transactions on Electronic Computers*, vol. EC-14, pp. 326-334, 1965

[6] A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*, Springer - Verlag, 2003.

[7] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, 1999

[8] C. - W. Hsu, C. - J. Lin, "A Comparison of Methods for Multi-class Support Vector Machines", *IEEE Transactions on Neural Networks*, 13(2), pp. 415-425, 2002

[9] T. Joachims, "Making Large-Scale Support Vector Machine Learning Practical", *Advances in Kernel Methods: Support Vector Learning*, B. Scholkopf, C. Burges, A. Smola, (Eds.), pp. 169-184, 1999

[10] B. Scholkopf, "Support Vector Learning", *Dissertation*, TU Berlin, 1997

[11] R. Stoean, D. Dumitrescu, C. Stoean, "Nonlinear Evolutionary Support Vector Machines. Application to Classification", *Studia Universitatis Babes-Bolyai, Seria Informatica*, 2005 (in press)

[12] R. Stoean, C. Stoean, M. Preuss, E. El-Darzi, D. Dumitrescu, "Evolutionary Support Vector Machines for Diabetes Mellitus Diagnosis", *IEEE IS* 2006 (submitted)

[13] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998

[14] J. Weston, C. Watkins, "Multi-class Support Vector Machines", *Technical Report*, CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science, 1998

Ruxandra Stoean, Cătălin Stoean
University of Craiova
Department of Computer Science
Address: 13, A. I. Cuza St., 200585, Craiova, Romania
E-mail: {ruxandra.stoean,catalin.stoean}@inf.ucv.ro

Mike Preuss
Dortmund University
Department of Computer Science
Address: 20, Joseph-von-Fraunhofer St., 44227,
Dortmund, Germany
E-mail: mike.preuss@cs.uni-dortmund.de

Dan Dumitrescu
Babes-Bolyai University, Cluj-Napoca
Department of Computer Science
Address: 1B, Mihail Kogalniceanu St., 400084,
Cluj-Napoca, Romania
E-mail: ddumitr@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 429-438

Generating JADE agents from SDL specifications

Florin Stoica

Abstract: SDL (Specification and Description Language) is an object-oriented, formal language defined by The International Telecommunications Union - Telecommunications Standardization Sector (ITU-T), applicable to the specification and implementation of distributed systems. The SDL tool market has expanded and changed significantly in the last years. The reason is, that it has become practical to use SDL for the (semi-) automatic generation of implementations. SDL tools can produce source code in programming languages (usually C/C++/Java) directly from a SDL specification and this code can be linked with a run time system to make products. The work presented in this paper consists on a tool developed to help the process of prototyping asynchronous, concurrent systems, such as agent - based systems. This tool is responsible for generating JADE agents automatically from SDL specifications. The generated code is a completely functional Java code.

Keywords: SDL, agents, systems modeling.

1 Introduction

It is widely accepted that the key to successfully developing a system is to produce a thorough system specification and design. This task requires a suitable specification language, satisfying the following needs: a well-defined set of concepts, unambiguous, clear, precise, and concise specifications, a thorough and accurate basis for analyzing specifications, a basis for determining the consistency of specifications, computer support for generating applications without the need for the traditional coding phase.
SDL has been defined to meet these demands. It is a graphical specification language that is both formal and object-oriented. The language is able to describe the structure, behaviour and data of real-time and distributed communicating systems with a mathematical rigor that eliminates ambiguities and guarantees system integrity. It has a graphic syntax that is extremely intuitive, providing quickly an overview of a system's structure and behaviour. The most important characteristic of SDL is its formality. The semantics behind each symbol and concept are precisely defined.
A system specification, in a broad sense, is the specification of both the behaviour and a set of general parameters of the system. However, SDL is intended to specify the behavioural aspects of a system. Thus, a SDL specification of a system is the description of its required behaviour.
Software agents are reactive - agents perceive their environment and respond to changes that occur. An agent can perform some or no actions, and possibly change its state, as a reaction to events, external (received from the environment) or internal (generated by its own previous actions).
Agents are the fundamental specification concept of SDL-2000 [3]. The behaviour of an agent is described as an extended finite state machine: when started, an agent executes its start transition and enters the first state. The reception of a signal triggers a transition from one state to a next state. In a state, an agent may execute actions (tasks). Actions can assign values to variable attributes of the agent, branch on values of expressions, call procedures, create new agent instances and send signals to other agents.
There are several, equally valid, ways of extending the basic finite state machine model into a model for SDL finite state machines. We select one of those models and formalize it in a definition of a reactive finite state machine. This model is also suited for FSM-driven behaviour of a JADE agent, implemented by FSMBehaviour class.
This model will help us to elaborate the mapping rules between SDL and JADE concepts, used by the SDL to JADE/Java code generation tool to translate an SDL specification into equivalent JADE code.

2 Reactive finite state machines

A reactive finite state machine is a tuple $(Q, \Sigma, \Delta, \delta, q_0, F)$ where
Q is a finite, non-empty set of symbols called states,
Σ is a set of symbols representing valid inputs,

Δ is a set of symbols representing valid outputs,
δ is the state transition function: $\delta : Q \times (\Sigma \cup \{none\}) \longrightarrow (Q \cup \{err\}) \times (\Delta \cup \{default\})$,
q_0 is an element of Q, the initial state,
$F \subseteq Q$ is the set of final states.
Elements from Σ will be called *signals*, and elements from Δ will be called *events*. In one reaction, a FSM associate a current state $p \in Q$ and an input signal $a \in \Sigma$ with a next state $q \in Q$ and an output event $b \in \Delta$, where $\delta(p,a) = (q,b)$.
The behavior of a FSM is more easily understood when this is represented graphically in the form of a state transition diagram. The control states are represented by circles, and the transition rules are specified as directed edges. Each transition is labeled by event from that triggers the transition. The arc without a source state denote then initial state of the system (state q_0).
During one reaction of the FSM, one transition is triggered, chosen from the set of admissible transitions (outgoing transitions from the current state), so that label of transition matches the terminating event of the current state. The FSM goes to the destination state of the triggered transition. Apparition of a terminating event for current state is conditionated by reception of one signal from Σ (leaving from a state could be done only if was received a signal), an exception being the special signal *none*, which induce a *spontaneous transition*.
If terminating event of the current state $q \in Q \setminus F$ is not explicit associated with an admissible transition, then:
• if exist the admissible transition labeled with *default*, this transition (called *implicit transition*) will be triggered
• else FSM goes in an inconsistent state, denoted through *err*.
In case if FSM arrive in a state $q \in F$, after completeness of activities from that state, execution of finite state machine is stopped.

3 Jade agents with FSM behaviours

JADE is a middleware that facilitates the development of multi-agent systems and applications conforming to FIPA standards for intelligent agents [11]. It includes:
• a **runtime environment** where JADE agents can "live" and that must be active on a given host before one or more agents can be executed on that host.
• a **library** of classes that programmers have to/can use (directly or by specializing them) to develop their agents.
• a suite of **graphical tools** that allows administrating and monitoring the activity of running agents.
The Agent class represents a common base class for user defined agents. Therefore, from the programmer's point of view, a JADE agent is simply an instance of a user defined Java class that extends the base Agent class, as shown in the code below:

```
import jade.core.Agent;
public class MyAgent extends Agent {
    protected void setup() {
    // Printout a welcome message
    System.out.println("Hello! The agent "+getAID().getName()+ " is ready!");
    }
}
```

The computational model of an agent is multitask, where tasks (or behaviours) are executed concurrently. Each functionality/service provided by an agent should be implemented as one or more behaviours. A scheduler, internal to the base Agent class and hidden to the programmer, automatically manages the scheduling of behaviours.
A behaviour represents a task that an agent can carry out and is implemented as an object of a class that extends jade.core.behaviours.Behaviour. In order to make an agent execute the task implemented by a behaviour object it is sufficient to add the behaviour to the agent by means of the addBehaviour() method of the Agent class.
Each class extending Behaviour must implement the action() method, that actually defines the operations to be performed when the behaviour is in execution and the done() method (returns a boolean value), that specifies whether or not a behaviour has completed and have to be removed from the pool of behaviours. Is important to notice that scheduling of behaviours in an agent is not pre-emptive (as for Java threads) but cooperative. This means that when a behaviour is scheduled for execution its action() method is called and runs until it returns. The termination value of a behaviour is returned by his onEnd() method [2].
The path of execution of the agent thread is showed in the following pseudocode:

```
void AgentLifeCycle() {
    setup();
    while (true) {
        if (was called doDelete()) {
```

```
                takeDown();
                return;
            }
            Behaviour b = getNextActiveBehaviourFromTheSchedulingQueue();
            b. action();
            if (b.done() returns true) {
                removeBehaviourFromTheSchedulingQueue (b);
                int terminationValueOfTheBehaviour = b.onEnd();
            }
        }
    }
```

Generic behaviours embeds a status and execute different operations depending on that status. They complete when a given condition is met:

```
    public class MyGenericBehaviour extends Behaviour {
        private int step = 0, stop = 2;
        public void action() {
                ...
                step++;
                ...
        }
        public boolean done() {
            return step == stop;
        }
    }
```

Behaviours work just like co-operative threads, but there is no stack to be saved. Therefore, the whole computation state must be maintained in instance variables of the Behaviour and its associated Agent. Following this idiom, agent behaviours can be described as finite state machines, keeping their whole state in their instance variables. When dealing with complex agent behaviours (as agent interaction protocols) using explicit state variables can be cumbersome; so JADE also supports a compositional technique to build more complex behaviours out of simpler ones. The abstract class JADE CompositeBehaviour provides the possibility of combining simple behaviours together (children) to create complex behaviours. The actual operations performed by executing this behaviour are not defined in the behaviour itself, but inside its children while the composite behaviour takes only. The CompositeBehaviour class only provides a common interface for children scheduling, but does not define any scheduling policy. This scheduling policy must be defined by subclasses. The FSMBehaviour is such a subclass that executes its children according to a Finite State Machine (FSM) defined by the user. More in details each child represents the activity to be performed within a state of the FSM and the user can define the transitions between the states of the FSM. When the child corresponding to state S_i completes, its termination value (as returned by the onEnd() method) is used to select the transition to fire and a new state S_j is reached. At next round the child corresponding to S_j will be executed. Some of the children of an FSMBehaviour can be registered as final states. The FSMBehaviour terminates after the completion of one of these children.
The following methods are needed in order to properly define a FSMBehaviour:

• public void **registerFirstState**(Behaviour state, java.lang.String name)
Is used to register a single Behaviour *state* as the initial state of the FSM with name *name*.

• public void **registerLastState**(Behaviour state, java.lang.String name)
Is called to register one or more Behaviours as the final states of the FSM.

• public void **registerState**(Behaviour state, java.lang.String name)
Register one or more Behaviours as the intermediate states of the FSM.

• public void **registerTransition**(java.lang.String *s*1, java.lang.String *s*2, int *event*)
For the state *s*1 of the FSM, register the transition to the state *s*2, fired by terminating event of the state *s*1 (the value of terminating event is returned by onEnd() method, called when leaving the state *s*1 - sub-behaviour *s*1 has completed).

• public void **registerDefaultTransition**(java.lang.String s1, java.lang.String s2)
This method is useful in order to register a default transition from a state to another state independently on the termination event of the source state.

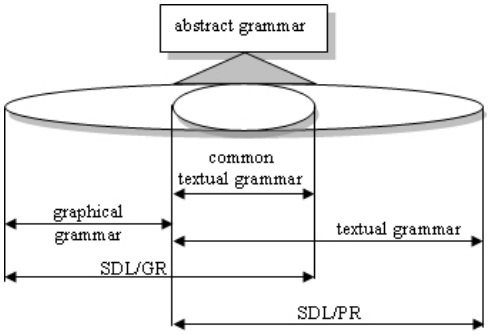


Figure 1: SDL grammars

4 SDL Systems

SDL Systems consist of a structure of communicating *Agents*. Each agent may have variables, procedures, a state machine and a structure. An agent is characterised by the signals it may receive from and send to other agents, and by the procedures that it may perform upon request. SDL provides the following kinds of diagrams [12]:
• ***Agent diagrams*** that describe the *properties of Agents*, in terms of variables, procedures, an *Agent state machine* and *contained Agents*;
• ***State diagrams*** that depict the behaviour of Agents in terms of *States* and state *Transitions*;
• ***Procedure diagrams*** that depict the behaviour of *Procedures*;
• ***Package diagrams*** that define types that can be used in other diagrams.
The behaviour of an agent is described as an Extended Finite State Machine: when started, an agent executes its start transition and enters the first state. The reception of a signal triggers a transition from one state to a next state. In transitions, an agent may execute actions (tasks). Actions can assign values to variable attributes of the agent, branch on values of expressions, call procedures, create new agent instances and send signals to other processes.
Communication by means of sending signals is asynchronous: the sending agent does no wait until the signal is handled by the receiver, and the receiving agent will keep signals in a queue until it reaches a state in which it is prepared to handle it.
SDL gives a choice of two different syntactic forms to use when representing a system: a Graphic Representation (SDL/GR), and a textual Phrase Representation (SDL/PR). As both are concrete representations of the same SDL, they are equivalent. In particular they are both equivalent to an abstract grammar for the corresponding concepts. A subset of SDL/PR is common with SDL/GR. This subset is called common textual grammar. Figure 1 shows the relationships between SDL/PR, SDL/GR, the concrete grammars and the abstract grammar [6].

5 Agent behaviour specification in SDL

This section introduces a simple example for an agent behaviour specification in SDL [10]. For building SDL specifications, has been used Cinderella SDL. Cinderella SDL is a CASE (Computer Aided Software Engineering) tool which is available from Cinderella (www.cinderella.dk).
Figure 2 shows the State diagram which describe behaviour of MyAgent agent as an Extended Finite State Machine (SDL/GR representation).
The SDL/PR representation of MyAgent is:

```
process MyAgent ;
signalset Inform, Result;
dcl i Integer :=0, j Integer :=0;
signal Inform(Integer), Result(Charstring);
start;
      task i:=i+1 ;
      nextstate State_A ;
state State_B ;
      input none;
      output Result('B') ;
```

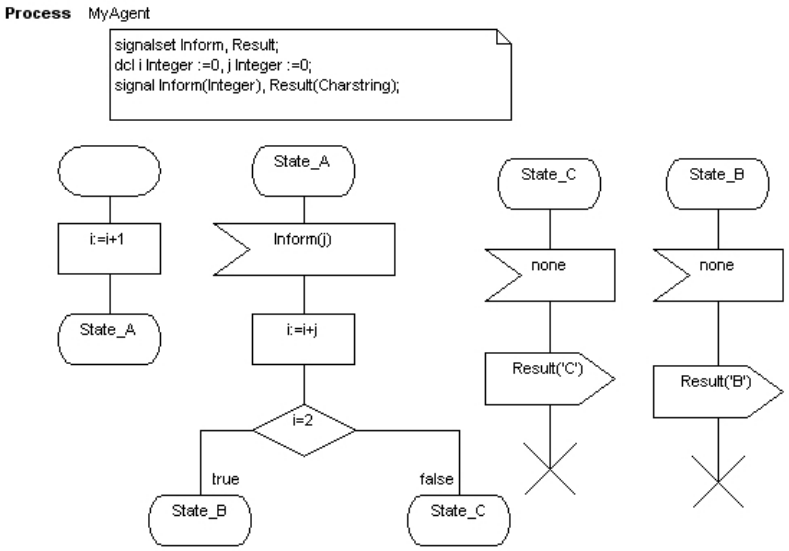


Figure 2: The SDL/GR representation of MyAgent (State diagram)

```
        stop;
    endstate;
    state State_C ;
        input none;
        output Result('C') ;
        stop;
    endstate;
    state State_A ;
        input Inform(j) ;
        task i:=i+j ;
        decision i=2 ;
            ( false ): nextstate State_C ;
            ( true ): nextstate State_B ;
        enddecision;
    endstate;
    endprocess;
```

The following table contains explanations about symbols used in SDL/GR representation of MyAgent:

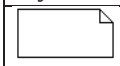
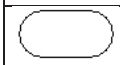
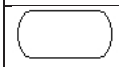
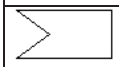
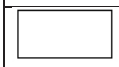
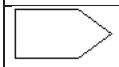
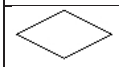
Symbol	Description
	A text symbol is a basic symbol which contains textual definitions (data types, signals etc.).
	The start symbol is used as start symbol for FSM
	The state symbol is used to represent a state of FSM
	The input symbol (receiving signals). A spontaneous signal symbol is an input symbol containing the text ***none*** (it induces spontaneous transitions)
	The task symbol (performing activities)
	The output symbol (sending signals)
	The decision symbol (branch execution)
	The stop symbol (execution of finite state machine is stopped).

Table 1: SDL graphical symbols

The executability of SDL does not only provide means for efficient implementations, but also for simulation in all stages of the design. In Cinderella SDL, we can simulate our specification. When simulation is started, an environment process is created from which stimuli can be sent to the system and which holds in its input queue the signals sent from the system. The *Specification explorer* is the basic feature for viewing and navigation in the SDL specification. The *Specification explorer* can be set up to show a large range of SDL related information in various forms.

6 Automatic Java Code Generation from SDL specifications

The work presented in this section consists on a translator which generate automaticaly Java code parsing a SDL/PR specification that should have been developed in a previous step (a SDL specification defined in the graphical format SDL/GR can be saved in the textual format SDL/PR using the File export command from Cinderella SDL). In Table 2 are defined mapping rules between SDL and JADE concepts. The SDL to JADE/Java code generation tool uses table 2 to convert the SDL specifications into equivalent JADE code. The tool reads a file that contains the SDL specifications in a textual form (SDL-PR) and, based on these specifications, produces the equivalent Java code for the behavioral description of the SDL system.

Typically, the procedure from requirements analysis to product implementation would involve the following steps:

• Collect the initial requirements

• Make the SDL diagrams (specifications) to a level where they can be analysed, simulated and checked for consistency with the system requirements analysis (this can be done in Cinderella SDL)

• When SDL design has proved consistent with the requirements, a code for the application can be generated.

In the following we present the tool responsible for generating JADE agents automatically from SDL design. This tool is based on the SDL parser developed by Michael Schmitt [7], which used ANTLR to build his own parser.

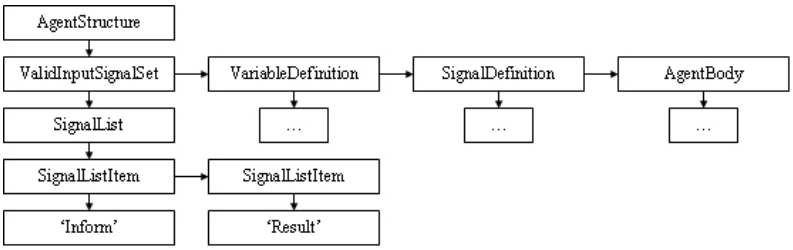


Figure 3: A fragment from AST generated for the agent MyAgent described above

SDL specification	**FSMBehaviour of a JADE agent**
Start symbol	The *setup()* method of the agent
State symbol	Child Behaviour registered as state of FSMBehaviour
Activities performed within a state	The *action()* method of child Behaviour associated with that state
Transition from current state to next state (*nextstate*)	The *done()* method of child Behaviour associated with current state returns ***true***
Stop symbol	The *doDelete()* method of class Agent, called from child Behaviour associated with SDL state within is reached the *stop* symbol; this Behaviour will be registered as final state in FSMBehaviour
Signal receiving	Receiving a JADE message
Receiving a ***none*** signal in a certain state	Execution of activities from *action()* method of child Behaviour associated with that state (and generation of a terminating event for that state), unconditionated by reception of a proper signal

Table 2: Mapping rules between SDL and JADE

ANTLR, ANother Tool for Language Recognition, is a tool that accepts grammatical language descriptions and generates programs that recognize sentences in those languages. ANTLR knows how to build recognizers that apply grammatical structure to three different kinds of input: (i) character streams, (ii) token streams, and (iii) two-dimensional trees structures. Naturally these correspond to lexers, parsers, and tree walkers. The syntax for specifying these grammars, the *meta-language*, is nearly identical in all cases. ANTLR knows how to generate recognizers in Java, C++, C# [1].

The parser developed by Michael Schmitt reflects the SDL-2000 standard correctly and produces the abstract syntax tree (AST) of a SDL/PR specification. Specification of the SDL grammars is provided in three files: SDLLexer.g, SDLParser.g and SDLTreeParser.g; these correspond to generated lexer, parser, and tree walker, respectively. If the *buildAST* flag in the ANTLR parser options section is set to true, the parser created by ANTLR will read the source langauge and create ANTLR abstract syntax trees in memory. The tree walker undertake the complete building of the abstract syntax tree. An abstract syntax tree (AST) captures the essential structure of the input in a tree form, while omitting unnecessary syntactic details. ASTs can be distinguished from concrete syntax trees by their omission of tree nodes to represent punctuation marks such as semi-colons to terminate statements or commas to separate function arguments. ASTs also omit tree nodes that represent unary productions in the grammar. Such information is directly represented in ASTs by the structure of the tree. Each node holds a token and pointers to its first child and next sibling:

With the implementation of the AST components complete, the subsequent phases of the translator has been implemented. The code generator provide Java/JADE code walking a derived AST, completed with actions which

are executed during tree walking. An action is a piece of code that run when a rule is matched. Actions can appear anywhere within a rule: before, during, or after a match.
The specification of the tree grammar is provided in the file SDLJADEParser.g.
Computational tasks of the tree walker, generated by ANTLR from the tree grammar, are:
• completeness of building of the abstract syntax tree, initiated by the parser;
• walking the final AST;
• generation of Java/JADE code, through execution of actions associated with tree nodes, during tree walking.
Invocation of the developed tree walker, and implicitly generation of JADE code is achieved in SDLMain.cpp:

```
// invocation of the parser
parser.sdlSpecification();
// reference to the AST builded by the parser
RefAST tree = parser.getAST();
jadeparser.setASTFactory( &factory );
// invocation of the "tree-walker" and code generation
jadeparser.sdlSpecification( tree );
// reference to the final AST
RefAST jadeTree = jadeparser.getAST();
```

The translator involve complex data structures. We decided to use the Standard Template Library (STL), which has C++ class templates for a wide range of basic building blocks of data structures, ranging from simple lists to associative arrays and hash tables.
For implementing the actions, the following variables/data structures were used:

```
static string firstState=""; //initial state
static string nextState=""; //next state
static string currentState=""; //current state
//associate each state with its current terminating code
//(event) which induce the transition;
// this represent the value of private variable _onEnd
//of the child behaviour which completes
typedef map<string, int, less<string>, allocator<int>>
TMapStateCode;
//associate each state of the FSMBehaviour with its child Behaviour
typedef map<string, string, less<string>, allocator<string>>
TMapStateBehaviour;
//associate each child Behaviour with its
//terminating condition from the done() method
typedef map<string, string, less<string>, allocator<string>>
TMapStateEnd;
// variable declarations
TMapStateCode mapStateCode;
TMapStateCode::iterator itStateCode; //iterator
TMapStateBehaviour mapStateBehaviour;
TMapStateBehaviour::iterator itStateBehaviour; //iterator
TMapStateEnd mapStateEnd;
TMapStateEnd::iterator itStateEnd; //iterator
```

The partial construction of the FSM which defines the scheduling policy for FSMBehaviour of the JADE agent is done when the tree walker recognize (match) the Nextstate node within AST:

```
nextstate :
#( Nextstate /*state*/ n:Name ( actualParameters )?(
      /*stateEntryPoint*/ Name )? )
{
if (firstState.empty()) {
      firstState = n→getText();
      mapStateBehaviour["start"].append(
            "fsm.registerFirstState(new Behaviour_");
      mapStateBehaviour["start"].append(firstState + "(), ");
      mapStateBehaviour["start"].append(toStr(firstState) + ");\n");
      if ((itStateBehaviour = mapStateBehaviour.find(firstState)) ==
            mapStateBehaviour.end()) {
```

```
            mapStateCode[firstState]=0;
            mapStateEnd[firstState]="";
            initStateBehaviour(firstState); //initialization of the
      }                                          //child Behaviour
}
else {
      nextState = n→getText();
      /** prepare to leave the current state **/
      mapStateCode[currentState]++;
      mapStateBehaviour[currentState].append("_onEnd =" +
            getStateCode(currentState)+";\n");
      if (mapStateEnd[currentState].empty()) {
            mapStateEnd[currentState].assign("_onEnd == " +
                  getStateCode(currentState));
      }
      else {
            mapStateEnd[currentState].append(" | | _onEnd == " +
                  getStateCode(currentState));
      }
      /** current state leaved **/
      mapStateBehaviour["start"].append("fsm.registerTransition(" +
            toStr(currentState) + ", " + toStr(nextState) + ", ");
      mapStateBehaviour["start"].append(getStateCode(currentState));
      mapStateBehaviour["start"].append(");\n");
} } ... ;
```

At finishing of the activities performed within a state, will be generated the methods done() and onEnd() of the associated child behaviour:

```
compositeStateApplication :
...
( end )?
{
mapStateBehaviour[currentState].append(
      "public boolean done() { \n");
mapStateEnd[currentState] = mapStateEnd[currentState].empty() ?
            "true" : mapStateEnd[currentState];
mapStateBehaviour[currentState].append("return (" +
            mapStateEnd[currentState]+");\n } \n");
mapStateBehaviour[currentState].append("public int onEnd() {\n");
mapStateBehaviour[currentState].append(
      "System.out.println(\"Behaviour \" + getBehaviourName() + \"
            completed.\");\n");
mapStateBehaviour[currentState].append(
      "System.out.println(\"Exit code: \" + _onEnd);\n");
mapStateBehaviour[currentState].append("return _onEnd; \n } \n }");
};
```

In the following is presented a fragment from generated code for the JADE agent, which contains the definition of the FSM which controls the FSMBehaviour:

```
      public class MyAgent extends Agent {
      //...
      protected void setup() {
            FSMBehaviour fsm = new FSMBehaviour(this) {
                  public int onEnd() {
                        System.out.println("FSM behaviour completed.");
                        myAgent.doDelete();
                        return super.onEnd();
                  }
            };
            //...
            fsm.registerFirstState(new Behaviour_State_A(), "State_A");
```

```
        fsm.registerLastState(new Behaviour_State_B(), "State_B");
        fsm.registerLastState(new Behaviour_State_C(), "State_C");
        fsm.registerTransition("State_A", "State_C", 2);
        fsm.registerTransition("State_A", "State_B", 3);
        addBehaviour(fsm);
    }
    //...
    }
```

After compilation, the generated agent MyAgent can be activated in the JADE platform:

```
java jade.Boot -gui myAgent:MyAgent
```

7 Summary and Conclusions

In this paper was presented implementation of a translator which generate automaticaly Java code parsing a SDL/PR specification of an agent, targeting the JADE platform. For this purpose, was ported on Windows platform (as a Microsoft Visual C++ 6.0 project) the SDL parser developed by Michael Schmitt on SuSE 8.1. This parser was extended with a tree parser grammar, SDLJADEParser.g. Because the project is based on ANTLR, was necessary to port on Microsoft Visual C++ the ANTLR runtime, resulting the library libantlr.lib, used to produce the generator executable.
In this stage of work, the generated code can be used as a prototype of a real agent-based application.

References

[1] ANTLR Reference manual, http://www.antlr.org

[2] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, *JADE programmer's guide*, http://jade.tilab.com

[3] R. Braek, A. Meisingset, *The ITU-T Languages in a Nutshell*, Telektronikk 4.2000

[4] G. Bucci, A. Fedeli, E. Vicario , *Specification and Simulation of Real Time Concurrent Systems Using Standard SDL Tools*, R. Reed (Ed.): SDL 2003, LNCS 2708, pp. 203-217, 2003

[5] J. Floch, R. Braek, *Using SDL for Modeling Behaviour Composition*, R. Reed (Ed.): SDL 2003, LNCS 2708, pp. 36-54, 2003

[6] ITU-T Recommendation Z.100, *Specification and description language (SDL)*, International Telecommunication Union (ITU), 2000

[7] M. Schmitt, *SDL-2000 parser/syntax checker*, http://www.teststep.org/

[8] B. Moller-Pedersen, *SDL Combined with UML*, Telektronikk 4.2000

[9] SDL Forum Society, http://www.sdl-forum.org/

[10] F. Stoica, *SDL executable specifications of agent-based systems*, The Proceedings of the International Economic Conference "25 Years of Higher Economic Education in Brasov", 2005, Ed. Infomarket, ISBN 973-8204-72-0

[11] The Foundation for Intelligent Physical Agents (FIPA), www.fipa.org

[12] SINTEF ICT, TIMe: The Integrated Method, http://www.sintef.no/time/

Florin Stoica
Lucian Blaga University of Sibiu
Department of Computer Science
Address: 12 Hipodromului St., Sibiu, Romania
E-mail: florin.stoica@ulbsibiu.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 439-444

A Programming Interface for Finding Relational Association Rules

Gabriela Şerban, Alina Câmpan, Istvan Gergely Czibula

Abstract: Association rule mining means searching attribute-value conditions that occur frequently together in a data set ([5]). Ordinal association rules are a particular type of rules and describe numerical orderings between attributes that commonly occur over a data set ([2]). We propose an extension of ordinal association rules, called relational association rules. Relational association rule mining can be used in solving problems from a variety of domains, such as: Data Cleaning, Natural Language Processing, Databases, HealthCare. In this paper we present a new programming interface for finding relational association rules, *RARI* (*Relational Association Rule Interface*). The main characteristic of this interface is its extensibility, meaning that new attribute types and new relations between attributes can be simply added. Consequently, using this interface, we can simply develop applications for finding relational association rules in different kinds of data. We report an experiment for finding relational association rules in medical data, developed using the designed interface.

Keywords: Relational association rules, Programming, Interface.

1 Introduction

The purpose of this paper is to present a standard interface for finding relational association rules that hold in a data set. The interface is meant to facilitate the development of software for finding relational association rules in different domains. For this issue, the entities, their characterizing attributes, the attribute types and the relations between attributes can be designed and implemented separately and then interconnected relatively easily, in a standard, uniform fashion.

1.1 Relational Association Rules

We extend the definition of ordinal association rules ([2]) towards *relational association rules*.

Definition 1. Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of entities (records in the relational model), where each record is a set of m attributes, $(a_1, \ldots, a_m)$. We denote by $\Phi(r_j, a_i)$ the value of attribute a_i for the entity r_j. Each attribute a_i takes values from a domain D_i, which contains ε (empty value, null). Between two domains D_i and D_j can be defined partial relations, such as: less or equal ($\leq$), equal (=), greater or equal ($\geq$), etc. We denote by $\mathcal{M}$ the set of all partial relations defined. An expression $(a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}) \Rightarrow (a_{i_1}\ \mu_1\, a_{i_2}\ \mu_2\, a_{i_3} \ldots \mu_{\ell-1}\, a_{i_\ell})$, where $\{a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}\} \subseteq \mathcal{A} = \{a_1, \ldots, a_m\}$, $a_{i_j} \neq a_{i_k}$, $j, k = 1..\ell,\ j \neq k$ and $\mu_i \in \mathcal{M}$, is an **relational association rule** if:

a) $a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}$ occur together (are non-empty) in s% of the n records and $s \geq s_{min}$ (given); we call s the **support** of the rule;

and

b) if we denote by $R' \subseteq R$ the set of records where $a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_\ell}$ occur together and $\Phi(r_j, a_{i_1})\ \mu_1\ \Phi(r_j, a_{i_2})\ \mu_2\ \Phi(r_j, a_{i_3})\ \ldots\ \mu_{\ell-1}\ \Phi(r_j, a_{i_\ell})$ is true for each record r_j din R', then $c = |R'|/|R| \geq c_{min}$ (given) holds; we call c the **confidence** of the rule.

In [2] is given a discovery algorithm for binary ordinal association rules (rules between two attributes). We developed in [3] an algorithm, called *DOAR* (Discovery of Ordinal Association Rules), that efficiently finds all ordinal association rules, of any length, that hold over a data set. We have proved that the proposed algorithm is correct and complete. This algorithm can be used for finding relational association rules, as well. For implementing the main functionality of our interface we have used the *DOAR* algorithm.

1.2 Aims of our approach

The aim of the proposed approach is to make an abstraction of the issue of finding relational association rules, ensuring a general approach, independent of the concrete entities, attributes and relations involved in the process of finding association rules.
In various domains clearly appears the necessity to find relational association rules that hold over different kinds of entity sets, characterized by different types of attributes and with various kinds of relations between attributes.
For example, in the field of Natural Language Processing we may need to find association rules between words. In this example, an entity is a context (set of words), the attributes are features characterizing the words, and the relations can be linguistical relations between words.
In relational databases, an entity is a record from a table, the attributes are table fields, and the relations are defined between attributes.
Another example in which finding relational association rules is useful is HealthCare. For example, for a given disease we assume that a set of patients are characterized by a set of symptoms. In this example, an entity is a patient, the attributes are symptoms and the relations are defined between symptoms.
There are other domains in which finding relational association rules can be used, for various types of entities, attributes and relations.
That is why, in this paper we propose an unitary approach for finding relational association rules in a data set, approach that is independent of the types of entities, attributes and relations.
The paper is structured as follows. Section 2 describes the functionalities provided by our interface. Section 3 presents its design. We then report an experiment developed using *RARI*.

2 The programming interface *RARI*

In this section we propose an API that allows a simple development of applications based on finding relational association rules. *RARI* ensures a uniform development for all these applications. It provides a hierarchy of classes that can be used for finding relational association rules.
The main advantage of the interface is that the user can simply define, depending on the current problem domain, new types of attributes and new types of relations between the attributes, while the process of finding relational association rules remains unchanged. The interface is realized in JDK 1.5, and is meant to facilitate software development for finding relational association rules in data.
There are six basic entities (objects): `Entity` (defining an entity from the data set), `Attribute` (that characterizes the entities), `AttributeType` (that represents the type of an attribute), `Relation` (defining partial orderings between attributes), `AssociationRule` (describing relational association rules in data) and `AssociationRuleGenerator` (responsible for extracting relational association rules that hold over the entity set, based on the attributes and relations already defined).
For designing the interface, we made an abstraction of the mechanism for generating relational association rules, in order the interface to be used for any kind of data (various types of entities, attributes and relations). Much more, the entities representing the data set are completely separated from the attributes that characterize them (an entity has to know nothing about the attribute, it has to know only about its behavior). Thus, we can easily change and add attributes characterizing the entities, and relations between attributes, without affecting the general process.
The `AssociationRuleGenerator` class is the main class of the interface and manages the process of finding relational association rules in the given data set in respect to the given attributes. This class provides an operation for finding association rules in data and implements the *DOAR* algorithm ([3]).
The interface also provides a class that models a set of entities (the data set from which we want to extract relational association rules), a class that models a set of relational association rules and a class (`RelationSet`) that manages the set of relations between the attributes (this class allows to manage dynamically the set of relations defined between attributes).
For using the interface in a specific task, for finding relational association rules, the user has only to:

- define specialized classes for the concrete attribute types (for example a class `AttributeInt` that extends the abstract class `AttributeType` if the attributes are integer attributes);
- define specialized classes for the concrete relations between attributes (for example a class `IntIntEqual` that extends the abstract class `Relation` if we want to describe the equality relation between two integers). For example, if the user wants to define a relation between the PSN of a person and its

birthdate (in order to verify the validity of the data), he can simply define a relation `StringDateRelation` and then provide this relation to the relation manager (`RelationSet`);
- construct the concrete entity set.

All other mechanisms needed for generating the rules are provided by the classes from the interface.
In the following we present the skeleton of an application for finding relational association rules. Let us assume that attributes are of integer type, and the only relation needed between attributes is "=".

- First, the user defines the class that defines the concrete attribute type.
  public class `AttributeInt` **extends** `Attribute{...}`
- Second, the user defines the class that defines the concrete relation between the attributes already defined.
  public class `IntIntEqual` **extends** `Relation{...}`

In the same manner as above, the user can define as many attribute types and relations between them as are needed in the current task.
In the application class, which initiates the process of finding relational association rules, the user has to: a) define a method that reads the data from an external device (file, database) and returns an `EntitySet`; b) add (register) the concrete relations defined above to the set of relations `RelationSet`.

```
public class Application {
   public Application(){
      // the manager of relations adds a new concrete relation to its set
      // of relations
      RelationSet.addRelation(new IntIntEqual());
      // the application provides a method to read the data and to construct
      // an EntitySet object
      EntitySet es = readData();
      // an instance of an object AssociationRuleGenerator is created from
      // the entity set created above
      AssociationRuleGenerator arg = new AssociationRuleGenerator(es);
      // the association rule generator generates the set of association
      // rules having a minimum support and confidence
      double minimumSupport = 0.9;
      double minimumConfidence = 0.8;
      AssociationRuleSet ars = arg.genAssociationRules(minimumSupport,
                               minimumConfidence);
      // the determined association rule set can be now processed
   }
}
```

Figure 1 shows a simplified UML diagram ([6]) of the interface, illustrating the hierarchy of classes. It is important to mention that all the classes provided by the interface remain unchanged in all applications for finding relational association rules. The Figure 1 illustrates the core of the interface; what is outside the core are the concrete classes that the user has to define, by extending the classes provided by the interface, in order to develop an relational association rules discovery application.

3 The Design of the Interface

The classes used for realizing the interface are the following:

- **`AttributeType`** is **`ABSTRACT`**. Models an abstract attribute type, identified by the type name. The class has operations for returning and modifying the type name, a method for verifying the equality of two types and an abstract method that creates a value of that type from a string. The concrete attribute types defined by the user of the interface, will extend the abstract attribute type, managing their concrete type and overwriting the abstract method from the abstract `AttributeType`;
- **`Attribute`**. Models an attribute characterizing the entity set, identified by a name, a type (an instance of the `AttributeType` class) and a value (that is an object).

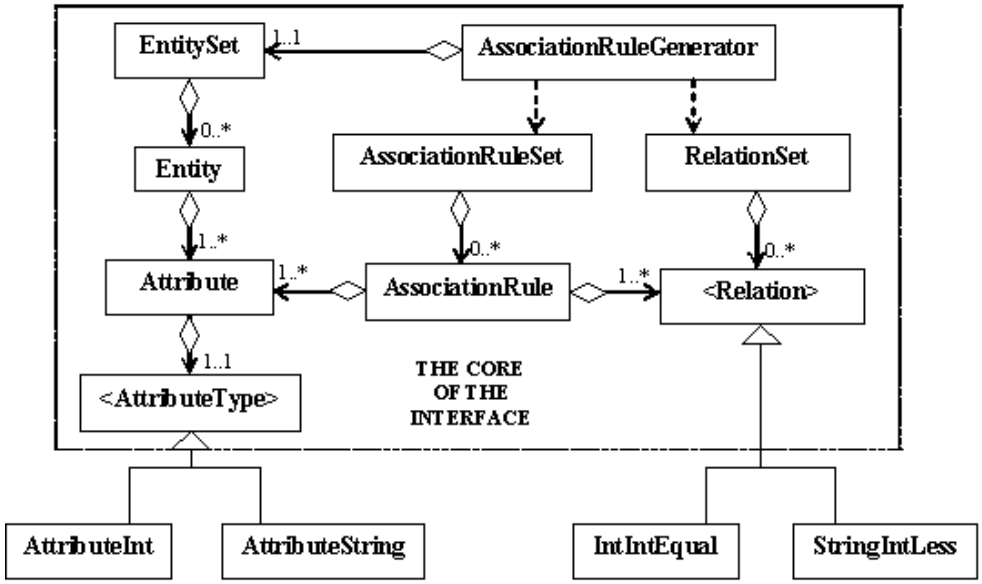


Figure 1: The diagram of the programming interface

- **`Entity`**. Models an entity from the data set, which consists in a list of attributes. The class has operations for managing the attributes: adding, removing and returning attributes from a given position, searching an attribute with a given name and type.
- **`EntitySet`**. Models a set of entities, which consists in a list of *Entity* objects. The class has operations for managing the set of entities: adding, removing, searching entities and a method that returns an iterator on the set.
- **`Relation`** is **`ABSTRACT`**. Models an abstract relation between two attribute types $Type_1$ and $Type_2$. The class has abstract operations for: returning $Type_1$ and $Type_2$, returning the name of the relation, verifying if two `Attributes` are in the given relation and for returning the converse of the relation.
- **`AssociationRule`**. Models an association rule, identified by a set of abstract attributes, a set of abstract relations, and characterized by its support and confidence. The main methods of this class are for: managing the attributes and relations from the association rule, setting and returning the support and the confidence of the rule.
- **`AssociationRuleSet`**. Models the structure of a set of association rules, which consists in a list of `AssociationRule` objects. The class has operations for managing the set of association rules: adding, removing, searching rules and a method that returns an iterator on the set.
- **`RelationSet`**. In our design this class models a repository (set) of relations, that allows the user to dynamically add relations between newly defined attribute types. The user can dynamically add new defined relations in this list, using a method *addRelation*. This class has methods for obtaining the relations for a given `Attribute`, for verifying if there exists a given `Relation` between two `Attributes`.
- **`AssociationRuleGenerator`**. Is the main class of the interface, that implements the process of finding relational association rules in a set of entities. It represents the *heart* of the interface, the uniform usage that all sets of entities, with their particular attributes and relations, are meant to conform to. The main method of the class is *generateAssociationRules*, that generates from the data set the relational association rules having a minimum given support and confidence, and returns an instance of the `AssociationRuleSet` class.

As it can be seen on Figure 1, there is a dependency relationship between the `AssociationRuleGenerator` and `RelationSet`, that allows the association rule generator to dynamically manage the relations added by the user, without affecting the main process of detecting rules.

4 Experimental Evaluation

In order to test *RARI*, we considered a HealthCare experiment that consists in detecting relational association rules between the symptoms of patients that have a cancer disease. We intend to use this method for medical diagnosis. The entities in this experiment are patients: each patient is identified by 9 attributes [1]. Each attribute represents a symptom in the cancer disease, and takes integer values between 1 and 10. In this experiment are 457 patients (entities). The data for this experiment was obtained from the website at "http://www.cormactech.com/neunet" ([4]). For this experiment, we have defined:

- `AttributeInt`, defining the integer attribute representing a patient's symptom in the cancer disease;
- `IntIntEqual`, `IntIntLess` and `IntIntGreater`, defining the possible orderings between two integer symptoms ($=$, $\leq$, $\geq$);
- a mechanism that read the data (for each patient, it reads the values of the symptoms) and creates an `EntitySet` object, in which a concrete entity is a patient.

Using the above defined classes, the application for detecting relational association rules between the symptoms was easily developed, based on the functionality provided by the interface.
By running *DOAR* with minimum support threshold of 90% and minimum confidence threshold of 80% we have obtained: 20 binary rules, 35 rules of length 3 and 19 rules of length 4. We intend to use the obtained results for assisting physicians in medical diagnosis.
As a conclusion of our experiments, we have to mention, from a programmer point of view, the advantages of using the interface proposed in this paper:

- is very simple to use;
- the effort for developing an application for relational association rule detection in a data set is reduced – we need to define only a few classes, the rest is provided by the interface;
- the user of the interface has to know nothing about the method of finding relational association rules, because it is provided by the interface;
- we can change the type of the entities or to add new attribute types and relations between attributes, while the interface remains unchanged.

5 Conclusions and Further Work

As a conclusion, we have developed a small framework that will help programmers to build, dynamically, their own applications for detecting relational association rules in different kinds of data, without dealing with the discovery technique (that remains unchanged and is provided by the interface). For a concrete application, the programmer has only to create, extending the classes defined by the interface, concrete classes for: the attribute types characterizing the entity set, the relations between the attributes and a mechanism for constructing the entity set from an input data (from a text file, a database, an XML file). So, the programmer's effort for developing an application is reduced.
Further works can be done in the following directions:

- how can the method for detecting relational association rules in medical data be used in medical diagnosis;
- how can the interface be generalized for adaptive association rule generation: new patients (entities) are added to the database or/and new symptoms (attributes) that characterize the patients are introduced.

References

[1] W. Wolberg, O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", *Proceedings of the National Academy of Sciences*, U.S.A., Volume 87, pp. 9193-9196, December 1990.

[2] A. Marcus, J. I. Maletic, K. -I. Lin, "Ordinal Association Rules for Error Identification in Data Sets", *CIKM 2001*, pp. 589-591, 2001.

[3] A. Campan, G. Serban, T. M. Truta, A. Marcus, “An Algorithm for the Discovery of Arbitrary Length Ordinal Association Rules”, submitted to *DMIN’06*.

[4] http://www.cormactech.com/neunet, “Discover the Patterns in Your Data”, CorMac Technologies Inc, Canada.

[5] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, The Morgan Kaufmann Series in Data Management Systems, 2001.

[6] http://www.omg.org/technology/documents/formal/uml.htm.

Gabriela Şerban, Alina Câmpan
“Babeş-Bolyai” University, Cluj-Napoca
Department of Computer Science
Address: 1, M.Kogalniceanu St., Cluj-Napoca
E-mail: {gabis,alina}@cs.ubbcluj.ro

Istvan Gergely Czibula
“InfoWorld” Cluj-Napoca
E-mail: czibula.istvan@infoworld.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 445-448

The Necessary Estimation of Space on Hard disk for the Implementation of Data Bases

Andy Ştefănescu

Abstract: The necessary estimation of space on hard disk can constitute a requirement in stage of projects the systems, pursuant to facts as the physical implementation of databases can be treatment by the configuration hardware current. Just but that not happened like this, still the designer must let estimates necessary of space on disk for the stockade databases in the eventuality in which it needs to buy new elements of hardware.

Keywords: file of types heap, hash, ISAM, arbor B* method.

1 Introduction

The estimation utilization disks depends in very big measure target of Database Systems and hardware elements used-up for support databases. By and large, the estimation is banked on the size of each record, on the number of records from relation. The last estimation must let us offers a maximum number. To settle down the potential sizes of databases in future, maybe it is necessary to take in the calculus the way in which will increase the relation and modification accordingly size disks.
Database is organized in one or many files, each file be constitute from one or more records, and each record be constitute from one or many fields. When an user require a record from database, the databases management system transposes the logical record in a physical record, find physical record in systems buffer from the capacity of primary stockade, using loading routine for access to file.
The physical is the unit of transfer between disk and main memory. By and large, a physical record is constitute from one or many logical record, with all as, sometimes, depending on size, a logical record can correspond of alone physical record. It's possibly as a large logical record contains more than a physical record. Sometimes, the terms: block and page maybe substitute the physical record.
Before illustrated this process it's needs to remember the fact as the designer establishes as objective an optimum organization of files, for each relation, based on follow types of file: Heap, hash, indexed sequential access method, arbor B*.

2 Optimal organization of files. A case study

A heap file represents most simple way to organizes the files of databases. The records are place into file, in same order was inserted. The main disadvantage of organization date in heap files is the fact as for find namely record is necessary a linear search. This presupposes the reading all pages from main-file to find demand record. This thing does founds from heap file with many pages to be relative slow.
Erased a record, before is necessary to find proper page, indication the record as be erase, following by writing the page backward on disk. The space of erased records can't be reworks. Pursuant, performance is deteriorated progressively, on measure are in progress the erases.
A INGRES page has 2048 bytes, from which 40 bytes are used-up like heading of page, and the remainder 2008 bytes are available for the stockage user dates. Total space necessary for a heap file can be a calculating according as follow:
rows_per_page = 2008/(large_row +2)
total_pages_heap = num_rows/ rows_per_page
Where: num_rows represent a record dimension.
For example, to calculate the size of table with 10000 records stocked like as heap file, will do:
rows_per_page = 2008/(111 +2) = 17
total_pages_heap = 10000/17 = 589
So, the table in cause will require the stockade of 589 INGRES pages as the heap file. In a system with size block of 512 bytes, the illustrates table will require 589*2048512 = 2356 blocks of disk

In **a hash file** don't needs as records to be write sequential way. Conversely, a hash function calculates the address for page in which will be stored the registration, based one or more fields from records. The records from hash file will appear as be distributed random in the available space from file.
The necessary for a hash file is caused like as the same algorithm used in case heap files. In addition, the number of pages must fit so that to take in consideration the fill-factor, which represents the percent from a page which will be used-up before it is considerate full.
In the INGRES system, the fill-factor established for hash file is 50%, if width record don't exceeds 1000 bytes - case in which the fill-factor is 100%. The formula becomes:
rows_per_page = (fill-factor *2008)/(large_row +2)
total_pages_hash = num_rows/ rows_per_page *(1/ fill-factor)
For example, to calculate table dimension with 10000 records stocked like as hash file with a fill-factor equal to 50%, will do:
rows_per_page = 0,5*2008/(111 +2) = 8
total_pages_heap = (10000/8)*2 = 2500
So, the table in cause will require the stockade of 2500 INGRES pages as the hash file with a fill-factor equal to 50%. In a system with size block of 512 bytes, the illustrates table will require 2500*2048/512=10000 blocks of disk
ISAM file contains sorted date with a primary index. This structure represents a compromise between a sequential pure file and an aleatory pure file. This in, because the records can be processed sequential or individual accessed used the value of search keys, what access the record among index key. A file sequential indexes represents a changeable structure who has:
- a primary stockade zone;
- a index or separate indexes;
- a upper exceeds

Indexed sequential access method ISAM from IBM use this structure, it is bound up with hardware quality.
Therewith organized file in this way accepts found based on exact corresponding keys. Because ISAM index is static, create when the file is creates, performances of ISAM file is deteriorated on measure what relation is updating.
In what look the necessary estimation of space on hard-disk, in case a ISAM file it is necessary to considerate the fill-factor and allocate a supplementary space for the ISAM index. In the INGRES system, the fill-factor is equal to 80%. The calculus of a sizes relation is accomplished after formula:
rows_per_pages = (fill-factor *2008)/(large_row +2)
empty = 2008 - (rows_per_pages * (large_row +2)
IF empty > (2048/(fill-factor*2048)) and (large_row $\leq$ empty
rows_per_pages = rows_per_pages + 1
total_pages_date = num_rows/rows_per_page
The calculus for sizes of index is accomplished after formulas:
key_per_page = 2008 / (large_key +2)
total_page_index/key_per_page + 1
Where: large_key = the size of key collumns
Thence, the global necessary space of a ISAM file is:
total_pages_isam = total_pages_date+total_pages_index
For example, to calculate table dimension with 10000 records stocked like as ISAM file with a fill-factor equal to 80%, will do:
rows_per_pages = 0.8*2008/(111+2) = **15**
total_pages_date = 10000/15 = **667**
key_per_page = 2008/(5+2) = **286**
total_pages_index = 770/286 + 1 = **4**
total_pages_isam = 667 + 4 = **671**
So, the table in cause will require the stockade of 671 INGRES pages as the ISAM file with a fill-factor equal to 80%. In a system with size block of 512 bytes, the illustrates table will require 671*2048/512=2684 blocks of disk.
An arbor B* file guy comprises the pages of indexes, branch pages intermediary, pages of date. The pages of date, of indexes and branch pages intermediary can use different factors for filling. In the INGRES system, fill-factors are equal to 80% for pages of date or indexes, respectively 70% for branch pages intermediary. The calculus for size of table is accomplished by formula:

```
rows_per_pages = (fill_date*2010)/(large_row +2)
empty = 2008 - (rows_per_pages * (large_row +2)
   IF empty > (2048-( fill_date *2048)) and (large_row≤empty
      rows_per_pages = rows_per_pages + 1
key_max = (1964/larg_key+6))-2
key_per_branchpages = key_max*fill_re_ branchpages
key_per_index = key_max *fill_index
num_ branchpages = (num_rows/key_per_ branchpages) + 1
num_pages_date=num_pages_date*(key_per_branchpages/rows_per_page)
+MODULO (num_rows/key_per_branchpages)/rows_per_page
num_ branchpages = 0
IF num_ branchpages≤key_per_index
THEN num_ branchpages = num_ branchpages/key_per_index
   ELSE num_pages_index = 0
      IF num_ branchpages > key_per_index
            THEN X = num_ branchpages
            DO
            X = X/key_per_index
            num_pages_index = num_pages_index + X
            WHILE (X > key_per_index)
      ENDIF
total_pages_arborb=num_pages_date+ num_ branchpages +
+num_pages_index+2
ENDIF
```

For example, to calculate the size of table with 10000 records stocked like as arbor B* file, indexed in one atributte, using the settle fill-factors, will do:
rows_per_pages = (0.8*2010)/(111+2) = **15** num_ branchpages == (10000/123)+1= **83**
key_max = (1964/(5+6))-2 = **176** num_pages_date = 83*(123/15)+3 = **750**
key_per_ branchpages = 176*0.7 = **123** num_pages_index = **0**
key_per_index = 176*0.8 = **140** total_pages_arborb = 750+83+0+2= **835**

3 Summary and Conclusions

This paper had like objective to do a comparative study for the estimation of necessary space on disk, in case of three types of files: heap, hash, ISAM.
Thence, the table taken as example will require store 835 INGRES pages like as arbor B* file, indexed one attribute, using the settle fill-factor. In the INGRES system with size block of 512 bytes, the table illustrates will require 835*2048/512 = 3340 blocks of disk.
In the next table is presented comparative, the needfulness of space on disk (in blocks of 512 bytes) for case in the records of table are increase

	10 000	**20 000**	**30 000**	**40 000**	**50 000**
Heap	2 356	4 708	7 060	9 412	11 768
Hash	10 000	20 000	30 000	40 000	50 000
ISAM	2 684	5 360	8 032	10 712	13 388
Arbor B*	3 340	6 592	9 844	13 100	16 352

This numerical examples put in evidence, in a table way, the best method for the of organization of data bases files.

References

[1] Connoly T., Begg C., Strachan A., *Database Systems - A practical approach to design, implementation and management*, Addison Wesley Limited, 1998

[2] Elmasri R.A., Navathe S., *Fundamentals of Database Systems*, New York, 1999

[3] Willits J., *Database design and Construction: Open learning course for Students and Information Managers*, Library association Publishing, New York, 2002

[4] Ştefănescu A., *Techniques to manipulate referential integrity in logic modeling of data*, The 10th Conference on Applied and Industrial Mathematics, Oradea, 2002

[5] Ştefănescu A., *Tehnici de prelucrare şi optimizare a interogărilor bazelor de date, în volumul: Realităţi şi perspective în dezvoltarea durabilă a economiei româneşti*, Editura Universitaria, Craiova, 2003, pag. 288.

Andy Ştefănescu
University of Craiova
Faculty of Economy and Business Administration
Address: 13, A.I. Cuza, Craiova, Romania
E-mail: andyboos73@yahoo.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 449-452

Using Data Warehouse for the Decisional Process of a Sustainable Firm

Laura Ştefănescu, Laura Ungureanu

Abstract: The paper presents some basic aspects regarding the use of data warehouse in the management activity: a data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.
At the same time this paper attempts to underline the reason for which data warehouse represent an urgent demand of the modern organisations to sustain the decisional process.

Keywords:sustainable firm, decisional process, data warehouse.

1 Introduction

In the developed countries where we can talk about real informational society and where the Its are actively present in the very day life, the data bases no matter the type (relational, distributed, object oriented, relational-object), are a common thing which every company constitute since the set up. The first thing an organisation starts with, of any type and size, is that of setting up it own data base, to take over an already existing one or to consult other data bases.
>From these set fourth so far we can draw the conclusion that the data bases have become common and take part from an ordinary work day in a informational large company. In any moment, the marketing employees can trace every order, the enrolment stage of any contract; they can offer the proper materials to the supply department. The engineers can programme best the high tide of production, they can find out in any moment where and in what quantity they can find a piece or a derived ensemble they can lead the technological process. The bookkeeping and the finances can be informational to reflect accurately, in value, the whole activity.

2 Arguments for using data warehouse by a sustainable firm

But to support the decisional process, as a means to achieve a competitive advantage, the attention will be concentrated on the means of using the operational data of the organisation. The problem of the decisional process deals with derives from the necessity that the data archives be transformed in a knowledge source, so that the decedent be presented a single integrated view on the organisation's data.
The new type of firm - based on knowledge - which is more and more present, must be a sustainable firm. The sustainable firm is that organisation which establishes economical objectives, social and ecological too, on long term, able to achieve them through strategic knowledge capitalisation, generating multiple positive consequences for stakeholders and its environment.
The sustainable firm needs data warehouse because these represent the solution to satisfy the demand of a system able to sustain the decisional process. The data warehouse represents a new concept which refers to the assistance of the decisional process of the new administration demands of organisations using the analysis of more information coming from different sources.
If, for instance, in a study which refers to the management of a sustainable firm, to draw up a viable strategic plan, we try to obtain some quick answers to a series of interrogatives which can elucidate the position on market.
Though being the manager of an ultramodern firm, completely informational, this one will encounter enough obstacles in rendering his own strategy. The informational system which he disposes rolls vital applications, pursuing data capture and data transforming to obtain necessary information for the operational activities.
In the above situation, the manager would be interested in knowing the clients potential who would want technique - town projects (for instance), to know the structure of the demand on types of services which the firm can provide etc. For that he asks the informational department a report, but this will last because the data must be taken from various subsystems, from different data bases, from different platforms, having to be brought in an intelligible form and, if possible, unitary.
Let's suppose that, eventually, the report is transmitted to him: a lot of tables from which the details are not relevant and the levels of summing are not sufficient. He will certainly want to try more variants of results grouping, but this could take many days.

Eventually, the manager realises that he should make a last years transactions analysis and should limit a series of measures depending on their evolution. But, the archive data are difficult to access and the formats in which they are differs from the present formats due to the fact that in the past few years the data bases were restructured several times, a new distributive system has been applied meanwhile etc.
In the end, the manager would like to group his data depending on the present situation, without asking the software developers to write applications, for a few months, that will be useless at the first change in the organisation of the informational system. But, on the other hand, spread sheet software is exceeded by these demands and an interrogation instrument in the complex structure of data bases cannot be imagined.
And still, the primary data exist and the external information is available. Why the present software system doesn't help him enough? Why the data cannot be transformed in information and relying on them he should make intelligent decisions? Because the system was conceived for office workers and not for managers, it is details - oriented, while the manager needs the general view. The present system wants to reflect as accurately as possible the current situation of the economical process, while the strategic leaders are interested in its evolution. The present system run vital applications for the organisation and does it well. Nothing should threaten its stability and security.
But, just for this thing to happen, he is after the capture of economical information, while the modern tendency towards the decentralisation of the decision would ask the dissemination of the information. Another drawback is that, the software system which he has an operational one, and the manager needs a software system. His hole construction looks for, mainly, the data processing (OLTP / on line transaction Processing), and the manager wants data analysis (OLAP - On line analytic Processing)
The apparition of the concept of data warehouse is justified by the limits of the online processing systems of transaction (OLTP), which can not provide very quickly and in the format required by the managers these information which they need. The OLTP type systems are specialised on types of problems, for instance: managing the production, making commercial transactions, having as main purpose to manipulate, quickly and rapidly, the data. To ensure the required performances and, sometimes, from historical or security reasons, they are not projected to work a co-operation with other systems.
Other reasons which indicate the OLTP type systems as being inadequate for the manager needs are linked to:
- Limiting the data manipulated by these systems to the current values necessary to fulfil their mission;
-Security aspects in the case in which the different data bases are distributed geographically.
If the problem is being analysed from the point of view of the software instrument which administrators the data base other complications can be observed. Nowadays SGBD of relation type are preponderant, that allow the effectuation of tens of thousand of transactions per minute. At the same time, relational SGBD allow to storage of some impressive data volumes in a non-redundant under the shape of tables which can be combined through certain operations, well founded mathematically, to obtain the information needed. The interrogation can be done flexibly using the SQL language. Despite all these advantages their using by a manager situated on the superior levels of leading hierarchy, needs knowledge and time which he doesn't possess in some cases.
To avoid all these short comings, an intermediary solution has been given by the Executive Information Systems - EIS. There are front-end type systems for OLTP type systems and have the mission to accomplish unit operations of the data received from OLTP which are stocked intermediary in the memory displays of EIS. The manager would better not formulate interrogation controls in a mysterious language for him, SQL, he has the possibility to choose from an already establish menu to obtain the necessary information.
Although EIS type solutions have represented an important step in the assistance of decisions data centred, they have some shortcomings due from the accomplishing conception itself. First of all data gathering from OLTP type systems remains a problem which is not simple from the technical point of view. Secondly, the EIS solution suffers from an inflexibility shown both in using it (the managers have to run several sequences of already established menus and nothing more or less and in supporting and development, situation in which the redesigning from the beginning can be necessary to consider supplementary requests of informing.
>From those shown and from the limits touched by the instruments of data bases administration systems we noticed that these determined the necessity of this new concept: data ware house and consequently, some new informational instruments of analytical processing online OLAP.

3 Scope and aims of data warehouse

The main purpose of data storing up is to integrate the general data from the whole organisation into one warehouse, from which the users can launch interrogations, produce reports and make analyses. The data warehouse represents

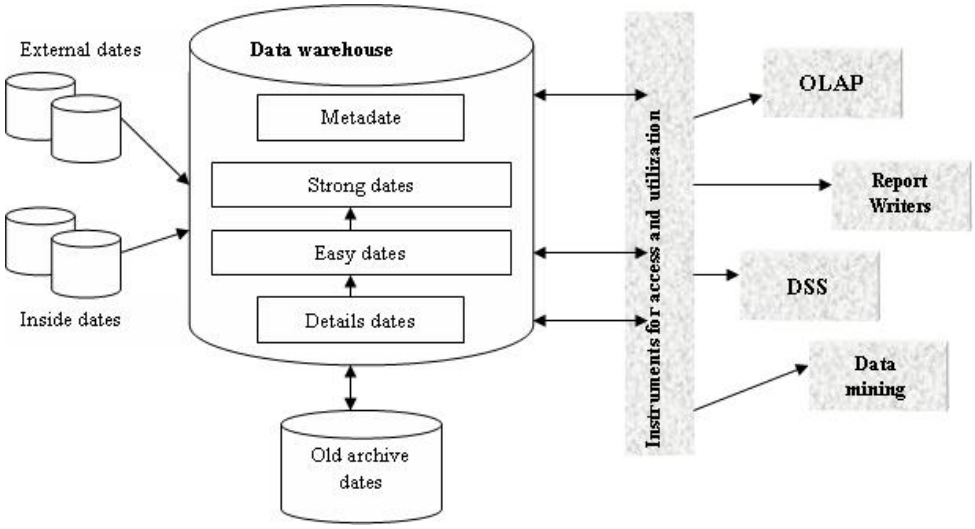


Figure 1: The architecture and the instruments to access at data warehouse

an environment of decisions sustenance, which takes over the data stored in the data bases, organises them and makes them available for the decisions organisms from the whole organisation.
The sources for data warehouse are: current operational data bases, archive old data bases, also external data bases. The structure of data ware has more types of date which are in correspondence with different information requires of users: details dates, aggregates dates, met dates, which describe the dates contented in data warehouse and the way in these are obtained and stocked.
Figure 1 put in evidence in a synthetically way, the main sources of date for warehouse, the structure of data warehouse but, also the instruments to access and utilisation of data warehouse. This instruments are ensure by the software associated to data warehouse, finding:
- Instruments necessary for users who need rapidly to punctual information's or report writers who transpose information into adequately forms;
- Instruments for assisted decisions which transpose information in charts, diagrams for analyse the trends, correlation and a suggests (OLAP, Data mining).
The main operations which are made on the data loaded from the data base are:
- Data loading from different sources: detecting the data of new interest placed in the source data bases and determining the way and the place of loading up;
- Data conversion from the original format into the one taken by the data warehouse;
- Data cleaning up, which comprises identification and correction functions of conversion errors and omissions completing;
- Data transforming through unit and summing up operations.

4 Conclusions

Briefly, the data warehouse is an administration technology and an analysis one of data which successfully implemented permits the managers to make a more substantial, more correct and more coherent analysis.
The OLAP type instruments are used to manipulate the data in a way that extends and makes flexible the functions and the way of operating of EIS type systems previously described. Intuitively, the OLAP functioning is suggested by "slice and dicing" type operations of the data base to allow the decision maker to find the information which allows him the finding of situations of interest or by pothesis verification. OLAP and the data warehouse are complementary. It is important to make the distinction between the concept of data warehouse and OLAP. While the data warehouse accumulates information having tactical character in a data base with a relational specialised capacity, to provide answers to questions such as: "Who ? and "Why ?", OLAP uses a multidimensional view of unit data to be able to answer to supplementary questions of these type: "Why ?" and "What if ?", typical for the systems for support decisions.

References

[1] Năstase FL., Năstase P., *Tehnologia aplicaţiilor WEB*, Economic Edition, Bucharest, 2003.

[2] Berson A., Smith S.J., *Data Warehousing, Data Minig and Olap*, Irwing McGraw-Hill, 1998.

[3] Airinei D., *Depozite de date*, Editura Polirom, Iaşi, 2002.

[4] Mallach E., *Decision Support and Data Warehouse Systems*, Irwing McGraw-Hill, 2000.

[5] Popescu I., *Decizia sau demersul antientropic de la cauză la efect*, Editura Universităţii, Lucian Blaga, Sibiu, 2005.

[6] Ştefănescu L., *Information Technologies Intends for Managerial Assistance*, The Second International Conference of Management and Industrial Engineering ”Sustainable Development Management”, 10-11 nov. Bucharest, 2005.

[7] Ungureanu L., *Modelling-Based Management*, The 10-th International Conference of Leadership and Management, pag. 183-189, Editura Academiei Fortelor Terestre, Sibiu, 2005.

[8] Watson H., *Data Warehousing Failures: Case Studies and Findings*, Journal of data warehousing, Vol.4, No.1, 2001.

Laura Ştefănescu, Laura Ungureanu
”Spiru Haret” University
Faculty of Accounting and Financial Management
Address: 4, Brazda lui Novac, Craiova, Romania
E-mail: laurastef73@yahoo.com, ungureanu@lycos.com

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 453-458

"The School in Your Pocket": Useful PocketPC Applications for Students

Horea Todoran, Adrian Sergiu Darabant

Abstract: Much smaller than laptops and still suitable for almost all kinds of applications, hand-held devices have the potential to rapidly become interesting tools for various daily activities. They can be successfully used in education by all participants (students, educators, administrative staff), if helpful applications are carefully designed and implemented. We propose a set of applications designed for PocketPCs, that allow educational actors to access their institution's information resources in a similar way they would do it from the desktop computers within the local area network. These applications could be integrated with the database system of a school or faculty, as well as with the available network services, thus creating "The School in Your Pocket" information system. It is the goal of this paper to briefly present the overall structure of the system and to stress on the applications dedicated to students.

Keywords: handhelds, mobile applications, databases

1 Introduction

Initially conceived as simple electronic organizers for address books and diaries, PDAs (Personal Digital Assistants, also known as palmtops or handheld PCs) eventually evolved into mini computers allowing users to carry out more complex, but still common activities like word processing, spreadsheet editing, multimedia presentation authoring. Most of the latest devices also offer wireless connectivity (WiFi, Bluetooth, infrared beam), thus facilitating data transfer across networks and other useful services, including web browsing, email, messaging. Moreover, current models include digital cameras, multimedia players (MP3), combine the function of a mobile phone and a PDA in one single unit, or even act as a GPS device to locate a destination. Consequently, handheld devices are able to accomplish virtually the whole range of tasks that a desktop computer or a laptop could perform. Additionally, they demonstrate a high degree of portability and mobility (about 10 times lighter than laptops or TabletPCs, fit into the jacket's pocket, can be hold into one hand and operated with the other, can run on the move), and lower power consumption (the battery life lasts roughly two times longer than for laptops). Last but not least, palmtops are cheaper than other mobile devices like TabletPCs or laptops (see [1] for a more detailed comparison of the various mobile devices) and very easy to use, therefore being called "equity computers" by some authors (Andrew Trotter quoted in [2]). These are the main reasons for us to believe that handheld computers have the potential to become more and more exploited in various fields in the near future,including education.

Nevertheless, there are also disadvantages when using PDAs, especially related to the small size of the screen, which limits the amount of information displayed or requires the intensive use of navigation bars. Data input is also more difficult than in the case of desktop computers or laptops, as PDA keybords and mouses are very small, if present. The navigation buttons are rather small and the stylus pens are narrow, thus requiring accurate operation on the screen pad. Palmtops have relatively limited storage capabilities, are quite difficult to upgrade and much less robust than TabletPCs, laptops and desktops ([1]). In order to overcome these limitations, software applications running on handhelds must be even more carefully designed than those for the other types of PCs. For instance, the Windows Mobile for PocketPC family includes scale down versions of the Microsoft Windows operating systems, very similar to the desktop versions, though being different in terms of minimal requirements (memory, processor speed) and visual elements (windows, menus, lists, buttons).
"The School in Your Pocket" is an information system that allows those involved in the educational process (students, teachers, administrative staff) to use PocketPCs for digitally interacting with each other. It is shaped according to the particular structure of the Romanian higher education system and it is intended to be implemented within Romanian faculties. However, most of the functionalities of the system could be easily adapted to elementary or high schools, colleges, or other educational institutions from within the country or from abroad. To the best of our knowledge, no similar project has yet been developed or working in the Romanian higher education. Taking into account the high number of desktop computers running MS Windows-like operating systems in Romania and the above mentioned versions of Windows Mobile, we decided to focus on PocketPCs rather than on PDAs running

PalmOS or other operating systems.

The next section of the paper will try to answer the question "Why do we need handhelds in higher education?". We will then briefly describe the overall structure and the main functionalities of the system. As a next step, we will focus on the set of applications dedicated to students, explaining their particularities and design issues. In the end, the conclusions and the future work plans are presented.

2 Benefits of using handhelds in higher education

There are four main categories of persons acting in the higher education that could mostly benefit from the use of handhelds: students, educators/trainers, researchers, and the administrative staff. Using palmtop devices provides all of them with *permanent access to information and services* from within the faculty LAN. The easiest solution for this to happen is the presence of a wireless network in the buildings of the institution, thus allowing all involved PDA owners to get access to the resources in the local area network and to the Internet via WiFi connections. This is especially important for students, who do not have theirown office and related facilities. From our personal experience, no matter how many computers installed in the labs, they are rarely enough for all students. The availability of information is also maintained when educational actors are far from the faculty LAN, but additional costs related to an Internet connection occur.

We will now shortly scrutinize the usefulness of using PDAs in the specific daily activities of each of the four target groups. According to their specific tasks, the members of the administrative staff could take advantage of PocketPCs especially by *accomplishing tasks on the move*. Examples include:

- inventory of books, furniture, computers, which are spread all over the campus or faculty buildings (controllers);
- take notes in various administrative meetings (secretaries);
- send electronic orders to or collect data from suppliers (procurers);
- remotely administrate computers and other equipments (network administrators);

Teachers (educators, trainers) could use PDAs for:

- *teaching activities* (e.g. presentation of educational content, record results of experiments, physical training, or outdoor activities, evaluate student work);
- *administrative tasks related to teaching* (e.g. take attendance to seminars, check lists with students who are allowed to take part to exams, grade students, distribute information, communicate with students);

For researchers, travelling to meetings or conferences requires a *high degree of mobility*, which can be achieved by means of PDAs. They are also helpful when conducting *experiments outdoor* (e.g. environmental, geological, sociological studies) or in spaces where desktop computers do not fit.

The main benefits for the students of a faculty using PocketPCs are related to *mobile learning*: students can do education specific activities (read or listen to courses, practice quizes, write notes, communicate with colleagues or teachers) wherever they are (waiting for busses, travelling by train or car, doing laundry), therefore saving an important amount of time.

Besides the benefits for individuals, the extended use of PDAs in an educational institution could bring important advantages for the institution itself. For instance, collecting data directly in digital form on handhelds, reduces the costs for paper, the time allocated to various activies, and the risk of errors.

3 Overall structure and functionalities of the system

The system is designed according to the client/server architecture, using a relational database server (Microsoft SQL 2000 Server) for the structured storage of information and a set of applications running on the client machine (the PocketPC of the user), developed using the Microsoft .NET Compact Framework.

3.1 Access to information

Each of the four groups of participants to the educational process has its own specific collection of tools, granting its members specific access rights to that parts of the database (projection) they are allowed to interact with. For example, students will be able to read the seminar attendance lists, but only professors are allowed to modify them, while the administrative staff will not have any tool to get access to this data.

Moreover, within each of the groups, a more restrictive projection on the database is done, according to the access rights of each member (authentication through username and password). A professor will be able to write grades only in the catalogues assigned to its courses, while a student would browse only his/her marks. In the end, we have a different view on the database for each educational actor.

3.2 Data synchronization between PocketPC and the database server

As already mentioned before, mobility is one of the most important gains of using PDAs for all educational actors. This means they are able to use up-to-date information virtually *whenever* and *wherever* they are located. In most of the cases, the user downloads the required information from the database server to the handheld device, processes a part of it offline on the PDA, and probably uploads it again in the end. The downloads/uploads should be as fast as possible, in order to save time and reduce communication costs, when the user is outside the faculty LAN.

The information stored in the faculty database is usually very dynamic, i.e. it changes often, even many times a day. Therefore, a *reliable data synchronization* between the PocketPC (client) and the database server is required.

There are three main options for synchronizing data between the Microsoft SQL server and the PocketPC. Using *XML web services*[1] is the only non-SQL Server synchronization, and allows developers to design and control all behavioural aspects. The use of the SOAP protocol and of the XML standard brings the advantage of interoperability with systems built on various software technologies. However, the method is slow, and therefore not suitable for large data volumes. The effort for code implementation is rather high, as the method is only a mechanism for transferring data between the client device and the database server, so there is no built-in support for data synchronization (conflict resolution) [4].

The other two methods, namelly *Remote Data Access (RDA)*[2] and *Merge Replication*[3] imply the installation of SQL server on the PocketPC and therefore require additional storage space on the mobile device, which sometimes could be a problem. RDA works according to the SQL Push/Pull method and is well suited to wireless transports. Compression is used to reduce the amount of transmitted data, which makes it viable also for slow connections (WWAN/dialup, WLAN). On the other hand, Merge Replication is based on the Publisher/Subscriber model, offers performant conflict resolution and ability to secure data synchronization. The main disadvantage with Merge Replication is probably the increased setup complexity. A more detailed comparison between the three methods, as well as a generic decision flowchart for choosing the synchronization architecture are presented in [5].

Taking into account the above mentioned characteristics of the three data synchronization methods, as well as the specific requirements of our system (low risk level for conflicts, intensive use of slow connection), we decided to adopt the Remote Data Access technology. Consequently, the overall structure of "The School in Your Pocket" system looks like in Figure 1 below.

4 Useful PocketPC applications for students

>From a managerial point of view, students are the "clients" of the educational institutions, as long as there is a strong correlation between the number of students and the size of the budget (from both subsidies and fees). Therefore, the level of satisfaction among the students is a significant indicator of the performance of each institution. In

[1] See http://msdn.microsoft.com/webservices/ for details
[2] Details at: http://msdn.microsoft.com/library/en-us/sqlce/htm/_lce_rda_intro_remote_data_access.asp
[3] See: http://msdn.microsoft.com/library/en-us/replsql/repltypes_6my6.asp

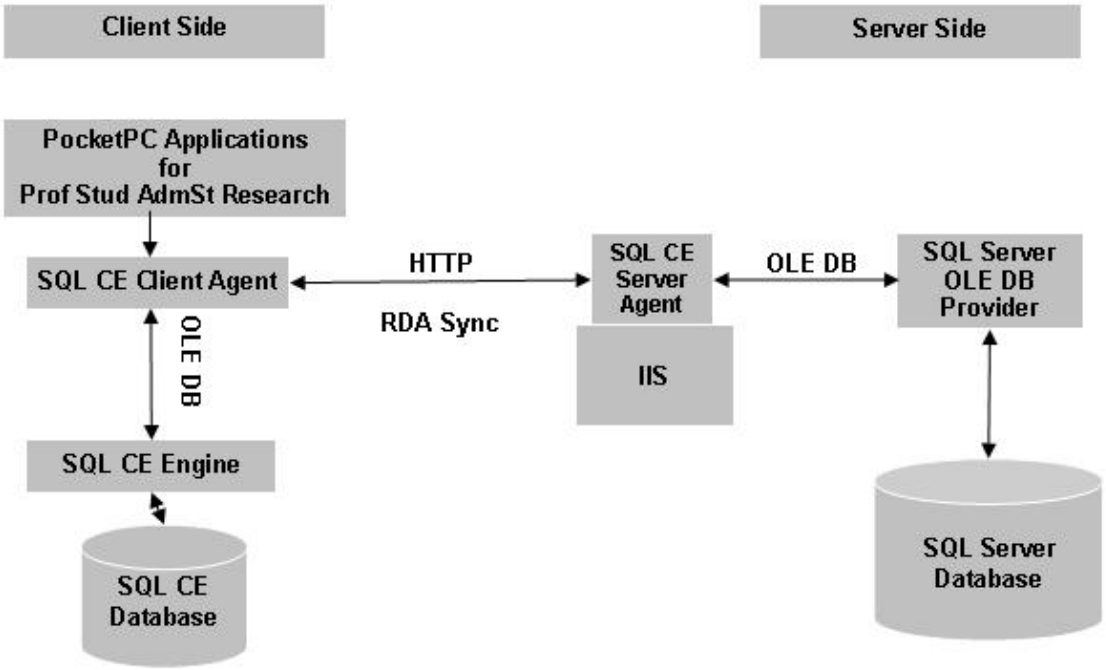


Figure 1: The overall structure of the system

this context, a central role is played by the availability of information and information-related services, especially those in digital form. The larger the variety of services offered by the educational institution, the higher the level of satisfaction among the students. Thus, a set of PocketPC applications for students, as we propose within "The School in Your Pocket" project has the potential to increase the level of satisfaction among the students, and, as a result, to improve the performance of the institution.

Most of the Romanian top universities have made important efforts in the last years to improve the digital communication between the student and the institution: computer networks have been built, websites have been designed and regularly updated, instruction has been offered in order to increase computer literacy. Nonetheless, we notice that the flow of information is mainly in one direction, from the institution toward the student. The latter has rarely the possibility to update its personal data in the system, or to submit an online form with its optional disciplines. In our opinion, a two-way student-faculty digital communication is a prerequisite for efficient distance and mobile learning, which are modern trends in the information society. Accordingly, our set of PocketPC applications for students tries to enhance the digital student-faculty interaction, facilitating the bi-directional flow of information.

In order to maximize the gains for students, the PocketPC applications have to be useful and carefully designed. Otherwise, they will become annoying and therefore rejected by users. Let us know briefly describe the general design requirements for PocketPC applications (independent from the application field), as well as the specific requirements for educational PocketPC applications dedicated to students.

4.1 General design requirements for PocketPC applications

In the introductory section of the present paper we have already outlined the three main limitations of current PDAs, in comparison to laptops and desktop computers: screen size, data input facilities, and memory/processing resources. All of them should be carefully considered when PocketPC applications are developed. An exhaustive list of general requirements for Windows Mobile-based PocketPC applications is given in [6]. Let us extract the most important ones, as minimal requirements for successful PocketPC applications:

- User Interface (UI)

- *Navigation Bar (NavBar)*: should continuously display the name of the topmost application; should not be hidden, as the Start Menu is the basic navigation mechanism of the PocketPC;
- *Soft Input Panel (SIP)*: the control should appear continuously in most of the cases, always in the bottom right corner of the screen; all text fields must be accessible with the SIP up; applications must resize when 80-pixel tall SIP appears or disappears at the bottom of the screen;
- *Common menus, buttons and functions*: menus on the Menu Bar must appear in the leftmost position; common menu items should appear in order (File, Edit, ViewĚ); common buttons should appear in order (New, Open, SaveĚ); common functions must appear order (Font, Font Size, **B**, *I*);
- *Help systems*: integrated with System TOC (authored in HTML format); only one entry point for Help (no button or item on the Menu Bar);
- *User settings*: applications must keep user settings (regional settings, themes);

- Functionality
  - *help* must be functional;
  - applications must not assume *external storage*;
  - graceful *shutdown and restoration of state* when starting are required;
  - *no shortcuts* in menus, tool-tips or help (no keyboard installed);

4.2 Specific requirements for applications dedicated to students

Students could use PocketPCs both for administrative and educational purposes, if useful applications are available. In our view, *useful administrative PocketPC applications for students* must at least ensure that students:

- can browse all their personal information connected to the faculty (grade records and exam results, situation related to fees, borrowed books) - the application should always provide the full name of the student, identification number within the institution, and the date of last update;
- can browse all public information within the faculty (timetables, news and announcements, structure of the faculty and contact data) - the application should always display the provider of the information, contact data, and the date of last update;
- don't get access to other students' personal data;
- can modify only the information they are responsible for, according to the access rights given by the system administrator (personal address or contact data, public or group announcements) - the application must track all the changes;
- can apply for optional courses, scholarships, participation to academic or social events, trainings, a place in student house - the system should correctly record the date/time of the application and the identity of the applicant;

Examples of *useful educational PocketPC applications for students* include:

- *topic specific compilations* (dictionaries, encyclopedias, lists of constants or chemical elements, historical events or geographical data, etc.) - the application should provide search capabilities to easily locate the required information;
- *testing systems* as tools for exam preparation - the application should insist on those types of exercises that don't require the extensive use of the keyboard (e.g. multiple choice, fill in the blanks, sequence building, matching, and so on);
- *assistive software* for impaired students (e.g. sound and speech libraries);

5 Conclusions and future work

The availability of useful PocketPC applications for students has the potential to improve their interaction with the faculty, in both administrative and educational purposes. Nevertheless, a set of general and specific requirements must be fulfilled (see the previous section) in order to develop applications that could gain students' acceptance. These applications might be integrated with similar software dedicated to the other groups of people involved in the educational process (professors, researchers, administrative staff). Taking into account the benefits of using handhelds in education (details in section 2), we propose "The School in Your Pocket" system as an unifying architecture for all the above mentioned applications (the overall structure of the system is presented in section 3). Future work will be dedicated to developing a framework for the creation of various types of exercises, that could be successfully used for distance education and mobile learning.

References

[1] K. Wood, "Introduction to Mobile Learning," *Ferl*, BECTa, 2003, [http://ferl.becta.org.uk].

[2] B. B. Ray, A. McFadden, S. Patterson, V. Wright, "Personal Digital Assistants in the Middle School Classroom: Lessons in Hand," *Meridian Palm Journal*, NC State University, Raleigh, NC, USA, Volume 4, Issue 2, 2001, [http://www.ncsu.edu/meridian/sum2001/palm/palm.pdf].

[3] D. Perry, *Handheld Computers (PDAs) in Schools*, Becta ICT Research Report, March 2003.

[4] M. D. Sutton, *Data Synchronization: Which Technology?*, Intel Software Network/DevX, September 2003, [http://www.devx.com/Intel/Article/17264].

[5] P. Dhingra, *SQL Mobile*, Microsoft Corporation, July 2005.

[6] ***, *Designed for Windows MobileŹ Software Application Handbook for Pocket PCs*, Microsoft Corporation, May 2004.

Horea Todoran, Adrian Sergiu Darabant
"Babes-Bolyai" University Cluj-Napoca
Address: 1, Mihail Kogalniceanu St., 400090 Cluj-Napoca, Romania
E-mail: htodoran@euro.ubbcluj.ro, dadi@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 459-464

Algebraic Model for the Counter Register Behaviour

Anca Vasilescu, Oana Georgescu

Abstract: We present an algebraic model based on Milner's SCCS for a specific structure of computer memory, namely the counter register. In this paper a concrete structure of the counter register is to be considered in order to achieve both the counting-down and the counting-up operations. Following the structural point of view, the target models are based on the behaviour of the internal components of the counter register, as follows: the T flip-flops, based on JK flip-flops, which are consequently based on the SR flip-flops. For each of these hardware components we define two different specifications and, both formally and automatically, prove that those specifications are equivalent. For the automatic verification of the target equivalences we use here the CWB-NC platform.

Keywords: SCCS process algebra, bisimulation equivalence, digital circuits, counter register

1 Introduction

The next results follow the contributions of the authors already obtained in [2], [5] concerning the use of an algebraic-based formal approach not only for studying the concurrent communicating processes, but for applying it in a different framework, namely the computer architecture and organization. In order to obtain our models, we shall use the Milner's process algebra SCCS, a synchronous calculus derived from CCS (*Calculus of Communicating Processes*)[4] and we shall combine the process algebra and the automata theory by using the CWB (*Concurrency WorkBench*) platform [7] for automatic verification of the target models.
In [5] it is mentioned that starting with the model of the flip-flops one may continue with the specification of any memory component behaviour, especially registers. Following this direction, we consider as prerequisites our results from [2], [5] concerning the *SR* flip-flops and the *JK* flip-flops, we define here appropriate models for *T* flip-flops and, finally, we specify and verify the algebraic models for the counter register.
This paper is structured as follows. Section 2 presents the preliminaries about digital circuits and SCCS language, so that in Section 3 we could develop the target algebraic model for the counter register. As a practical dimension, we consider a concrete internal structure of a counter register so that, depending on the input values of the circuit, the given register may operate either like a counting-down register or like a counting-up register.

2 Preliminaries

2.1 Flip-flops. Registers. Counter register

A *flip-flop* is a sequential circuit, a binary cell capable of storing one bit of information. It has two outputs, one for the normal value and one for the complement value of the bit stored in it. A flip-flop maintains a binary state until it is directed by a clock pulse to change that state. The difference among various types of flip-flops is the number of inputs and the manner in which the inputs affect the binary state. The most common types of flip-flops are: *SR* flip-flop, *D* flip-flop, *JK* flip-flop and *T* flip-flop.
A *register* is a group of flip-flops with each flip-flop capable of storing one bit of information. In this paper we refer to a *two-bit register* that has a group of two flip-flops and consequently it is capable of storing any binary information of two bits. In addition to the flip-flops, a register may have combinational gates that perform certain data-processing tasks. The flip-flops hold the binary information and the gates control when and how new information is transferred into the register.
A register that goes through a predetermined sequence of states upon the application of the clock pulses is called a *counter register*. Counters are found in almost all equipment containing digital logic. They are used for counting the number of occurrences of an event and they are useful for generating timing signals to control the sequence of operations in digital computers. A *two-bit binary counter* follows a sequence of states according to the binary count of two bits, from 0 to 2^2-1. The design of binary counters can be carried out from a direct inspection of the sequence of states that the register must undergo to achieve a straight binary count or a reverse binary count. *Synchronous binary counters* have a regular pattern, namely the clock inputs of all flip-flops receive the common clock.

2.2 Process algebra SCCS

The process algebra SCCS, namely *Synchronous Calculus of Communicating Systems* is derived from CCS [3], [4] especially for achieving the synchronous interaction in the framework of modelling the concurrent communicating processes. Both in CCS and in SCCS, processes are built from a set of atomic actions A. Denoting the set of labels for these actions by Λ, a CCS action is either (1) a *name* or an input on $a \in \Lambda$ denoted by a, (2) a *coname* or an output on $a \in \Lambda$ denoted by $\overline{a}$ or (3) an internal on $a \in \Lambda$ denoted by τ. In SCCS the *names* together with the *conames* are called the *particulate actions*, while an *action* $\alpha \in \Lambda^*$ can be expressed uniquely (up to order) as a finite product $a_1^{z_1} a_2^{z_2} \ldots$ (with $z_i \neq 0$) of powers of names. Note the usual convention that $a^{-n} = \overline{a}^n$ and that the action 1 in SCCS is the action τ from CCS and it is identified in SCCS with the empty product. A *process* P is defined with the syntax:

P	::=	nil	termination
	|	α:P	prefixing
	|	P+P	external choice
	|	P $\times$ P	product, synchronous composition
	|	P$\backslash L$	restriction, $L \subseteq A \cup \overline{A}$
	|	P$[f]$	relabelling with the morphism $f : A \cup \overline{A} \to A \cup \overline{A}$

The agent P$\backslash L$ is forced to not execute the actions from the set L, except as the internal actions. The operational semantics for SCCS is given via inference rules that define the transition available to SCSS processes. Combining the product and the restriction, SCCS calculus defines the synchronous interaction as a multi-way synchronization among processes.

3 Algebraic models

For each of the target circuits, we define two specifications as follows: a high-level specification Spec which is based on the definition of the circuit and a lower-level specification Impl which is based on the behaviour of that given circuit. As a demonstration technique, we start with the specifications Spec and Impl and then we apply a set of SCCS-based algebraic laws in order to formally prove that Impl is correct with respect to the Spec. This correctness proof is based on the *bisimulation congruence*, the appropriate equivalence in the theory of concurrent communicating processes. Implicitly, this result of bisimilarity shows that the behaviour follows the definition of the given system and, on the other hand, it is a guarantee of using that model in other complex circuits.

3.1 T flip-flop algebraic model

In order to obtain the model of the T flip-flop, we consider as prerequisites the appropriate processes already defined in [5] for modelling the JK flip-flop behaviour. We can summarize that results with the agents

$$\mathrm{SpecJK}(m,n,c) = \sum_{J,K \in \{0,1\}} (\mathrm{CLK}_c \alpha_J \beta_K \overline{\gamma}_m \overline{\delta}_n : \mathrm{StepJK}(m,n,J,K,c)) \tag{1}$$

where J and K are the input values, (m,n) $\in \{(0,1),(1,0)\}$ are the binary combinations on the outputs and c is the clock signal of the circuit. Note the convention that we use the SCCS names for the inputs of the circuits and the SCCS conames for the outputs. The current value of a variable like α_J shows that the circuit line α is carrying the logic value J.

By definition, a T (Toggle) flip-flop is obtained from a JK type with respect to the logic diagram represented in Figure 1. Consequently, when $T = 0$ a clock transition does not change the state of the flip-flop and when $T = 1$ a clock transition complements the state of the flip-flop.

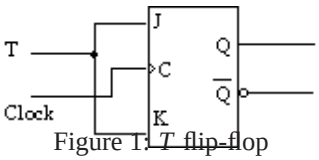


Figure 1: T flip-flop

For the current value T on the input we specify the entrance-level by the SCCS agents

$$\mathrm{InT}(T) = \xi_T \overline{\alpha}_J \overline{\beta}_K : \mathrm{InT}(T) \text{ and } \mathrm{SpecInT} = \sum_{T \in \{0,1\}} \mathrm{InT}(T) \tag{2}$$

where the evaluations are $J = T$ and $K = T$.
For the low-level specification of the system we model the entrance-level by the agents

$$\text{NODE} = in\overline{Up}\,\overline{Down} : \text{NODE and ImpInT} = \sum_{T\in\{0,1\}} \text{NODE}[\Phi_T] \tag{3}$$

where the morphism Φ_T is defined by the rellabeling pairs $in \mapsto \xi_T$, $Up \mapsto \alpha_J$ and $Down \mapsto \beta_K$ with $J = T$ and $K = T$.
Considering the JK flip-flop specifications from (1) and with respect to the meaning of the variables we propose the specifications Spec and Impl for the T flip-flop with the agents

$$\text{SpecT}(m,n,c) = (\text{SpecInT} \times \text{SpecJK}(m,n,c))\backslash\{\alpha,\beta\} \tag{4}$$

$$\text{ImpT}(m,n,c) = (\text{ImpInT} \times \text{ImpJK}(m,n,c))\backslash\{\alpha,\beta\} \tag{5}$$

Proposition 1. *The previous agents SpecT(m,n,c) and ImpT(m,n,c) for (m,n)* $\in \{(0,1),(1,0)\}$ *and* $c \in \{0,1\}$ *are bisimulation equivalent.*

Proof. The bisimulation relation '$\sim$' is a congruence over the class $\mathcal{P}$ of processes [4]. Besides, it is already established in [5] that $\text{SpecJK}(m,n,c) \sim \text{ImpJK}(m,n,c)$. Comparing the definitions (4) and (5) it follows that we only need to prove that SpecInT $\sim$ ImpInT. We shall prove that these target agents are solutions of the same equation. This together with the result of unique solution up to bisimilarity from [4] provide our target bisimulation equivalence.
For $J = T$ and $K = T$ we have

$$\text{NODE}[\Phi_T] = \xi_t\overline{\alpha}_J\overline{\beta}_K : \text{NODE}[\Phi_T] \tag{6}$$

Let consider the equation $X = \xi_T\overline{\alpha}_J\overline{\beta}_K : X$, where X is a guarded variable in the right-hand expression. The relations (2) and (6) show that the agents $\text{InT}(T)$ and NODE $[\Phi_T]$ are solutions of the previous equation. It follows that we have $\text{InT}(T)\sim\text{NODE}[\Phi_T]$. Because the bisimulation relation is a congruence, comparing the definitions (2) and (3) we have successively SpecInT $\sim$ ImpInT and SpecT(m,n,c) $\sim$ ImpT(m,n,c), as required. ■

If we present in details the models of the internal intercommunicating components, we obtain the next equation for the T flip-flop behaviour

$$\text{SpecT}(m,n,c) = \sum_{T\in\{0,1\}} (\text{CLK}_c\xi_T\overline{\gamma}_m\overline{\delta}_n : \text{SpecStepT}(m,n,T,c)) \tag{7}$$

where the agent SpecStepT specifies the behaviour of the T flip-flop after the first synchronized values interchange.

3.2 Algebraic model for the counter register

For the scope of this paper, we consider a concrete counter register based on a specific combination of interconnected digital components represented in Figure 2. A close version of this circuit is also treated in [2] as

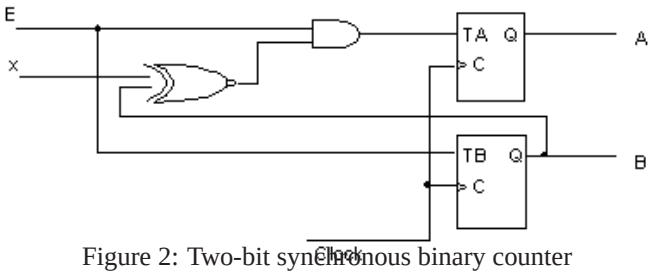


Figure 2: Two-bit synchronous binary counter

the diagrammatic solution of a concrete Boolean problem modelling the classical concurrent producer-consumer buffer. In this paper we are concerned about the algebraic model of the counter register as presented in Figure 2. This circuit consists of an input combinational level and a sequential level, the latter based on two synchronized T flip-flops. Hence, the above circuit operates as follows. If the count enable E is 0, both the TA and TB inputs are maintained at 0 and the output of the counter does not change. When the counter is enabled and the clock goes

through a positive transition, the operation of the counter depends on the value of the input x, as follows: while $x = 0$ the register operates like a counting-down counter and while $x = 1$ the register operates like a counting-up counter.
In this model, the bit B is the low-order bit and the bit A is the high-order bit. Hence, the predeterminated sequence of states for the content AB of the register is $00 \to 11 \to 10 \to 01 \to 00$, and so on, for the counting-down operation and it is $00 \to 01 \to 10 \to 11 \to 00$, and so on, for the counting-up operation.
For the Boolean evaluation $p = E$ AND (x NXOR B) and for $B \in \{0,1\}$, let define the high-level specifications for the combinational entrance level by the agents

$$\text{SpecLevel1}(B) = \sum_{E,x\in\{0,1\}} (\alpha_E \beta_x \overline{andout}_p \overline{Down}_E : \text{SpecLevel1}(B)) \tag{8}$$

In order to obtain the lower-level specifications, we first have to model the internal structure of logic gates, namely
(a) the distribution signal points using the agents $\text{NODE2} = \sum_{k\in\{0,1\}} (in_k \overline{up}_k \overline{down}_k : \text{NODE2})$ and
$\text{NodeE} = \text{NODE2}[\Phi]$ where the morphism Φ is defined by the rellabeling pairs $in_k \mapsto \alpha_E$, $up_k \mapsto Up_E$ and $down_k \mapsto Down_E$ with $E = k$;
(b) the NXOR gate using the agents $\text{NXOR}(B) = \sum_{i\in\{0,1\}} (nxorin_i \overline{nxorout}_r : \text{NXOR}(B))$ where the Boolean evaluation is $r = i$ NXOR B and the agents $\text{NXORx}(B) = \text{NXOR}(B)[\Psi_x]$ where the morphism Ψ_x is defined by the rellabeling pair $nxorin_i \mapsto \beta_x$, with $x = i$;
(c) the AND gate using the agents $\text{AND} = \sum_{i,j\in\{0,1\}} (andin1_i andin2_j \overline{andout}_p : \text{AND})$ where the Boolean evaluation is $p = i$ AND j and the agents $\text{ANDE} = \text{AND}[\Theta]$ where the morphism Θ is defined by the relabelling pairs $andin1_i \mapsto Up_E$, $andin2_j \mapsto nxorout_r$ with $E = i$ and $r = j$.
Finally, let $\text{ComLevel1} = \{Up, nxorout\}$ be the set of communicating actions for the appropriate processes involved in the next definition of the low-level specification of the entrance circuit:

$$\text{ImpLevel1}(B) = (\text{NodeE} \times \text{NXORx}(B) \times \text{ANDE}) \backslash \text{ComLevel1} \tag{9}$$

Proposition 2. *The previous agents SpecLevel1(B) and ImpLevel1(B) for $B \in \{0,1\}$ are bisimulation equivalent.*

Proof. We evaluate the relabelled agents as follows
$\text{NodeE} = \sum_{E\in\{0,1\}} (\alpha_E \overline{Up}_E \overline{Down}_E : \text{NodeE})$
$\text{NXORx}(B) = \sum_{x\in\{0,1\}} (\beta_x \overline{nxorout}_r : \text{NXORx}(B))$, with $r = x$ NXOR B
$\text{ANDE} = \sum_{E,r\in\{0,1\}} (Up_E nxorout_r \overline{andout}_p : \text{ANDE})$, with $p = E$ AND r.
Hence, considering the definition (9) the low-level specification can be largely explained by

$$\begin{aligned} \text{ImpLevel1}(B) &= (\textstyle\sum_{E\in\{0,1\}} (\alpha_E \overline{Up}_E \overline{Down}_E : \text{NodeE}) \times \sum_{x\in\{0,1\}} (\beta_x \overline{nxorout}_r : \text{NXORx}(B)) \times \\ &\times \textstyle\sum_{E,r\in\{0,1\}} (Up_E nxorout_r \overline{andout}_p : \text{ANDE})) \backslash \text{ComLevel1} = \\ &= \textstyle\sum_{E,x\in\{0,1\}} (\alpha_E \overline{Down}_E \beta_x \overline{andout}_p : \text{ImpLevel1}(B)) \end{aligned}$$

with $p = E$ AND r and $r = x$ NXOR B. We finally have $p = E$ AND (x NXOR B) and because an SCCS multi-particulate action is unique, up to order of the internal particulate actions, the agents are

$$\text{ImpLevel1}(B) = \sum_{E,x\in\{0,1\}} (\alpha_E \beta_x \overline{andout}_p \overline{Down}_E : \text{ImpLevel1}(B)) \tag{10}$$

If we compare now the expressions (8) and (10) of the two specifications SpecLevel1 and ImpLevel1, for each Boolean value $B \in \{0,1\}$, it follows that both agents are solutions of the same equation. According to the result of the unique solution up to bisimilarity [4] for the appropriate equation, we conclude that $\text{SpecLevel1}(B) \sim \text{ImpLevel1}(B)$, for each $B \in \{0,1\}$. ∎
In order to obtain the specifications of the counter register, the sequential level of the circuit has to be added to the above model. According to the previous Figure 2, this second level consists of the clock input and the two T flip-flops. In the next part of this section, we present in details the high-level specification for one step operation of the counter register.
For specifying the entrance combinational level of the counter register we define the agents $\text{SpecCLC}(B) = \text{SpecLevel1}(B)[\Phi]$ with $B \in \{0,1\}$ and the morphism Φ defined by the relabelling pairs $andout_p \mapsto TA_p$ and $Down_E \mapsto TB_E$.
For modelling the clock signal we define the agents $\text{NODE2}(k) = in_k \overline{up}_k \overline{down}_k : \text{NODE2}(k)$ and $\text{SpecCLK}(c) = \text{NODE2}(c)[\Psi_c]$ with $c \in \{0,1\}$ and the morphism Ψ_c defined by the relabelling pairs $in_k \mapsto CLK_c$, $up_k \mapsto CLKA_c$ and $down_k \mapsto CLKB_c$ with $c = k$.

We also propose the agents SpecT(A,c) and SpecT(B,c) for modelling the behaviour of the two T flip-flops by SpecT(A,c) = SpecT$(m,n,c)[\Theta_A^{(1)}]$ where the morphism $\Theta_A^{(1)}$ is defined by the relabelling pairs $\xi_T \mapsto TA_p$, $CLK_c \mapsto CLKA_c$, $\gamma_m \mapsto nil$ and $\delta_n \mapsto \delta_A^{(1)}$ with $p = T$ and SpecT(B,c) = SpecT$(m,n,c)[\Theta_B^{(0)}]$ where the morphism $\Theta_B^{(0)}$ is defined by the relabelling pairs $\xi_T \mapsto TB_E$, $CLK_c \mapsto CLKB_c$, $\gamma_m \mapsto nil$ and $\delta_n \mapsto \delta_A^{(0)}$ with $E = T$.
Note that the values of the variables TA and TB represent the entrance content of the register and the calculated values of the variables $\delta^{(1)}$ and respectively $\delta^{(0)}$ represent the next content of the register, namely after one step of operation.
Let ComCR(c) = $\{TA, TB, CLKA_c, CLKB_c\}$ be the set of communicating actions involved in the high-level specification of the counter register defined by the agent

$$\text{SpecCR}(A,B,c) = (\text{SpecCLC}(B) \times \text{SpecCLK}(c) \times \text{SpecT}(A,c) \times \text{SpecT}(B,c))\backslash\text{ComCR}(c) \tag{11}$$

In the same manner, if we consider the appropriate low-level specifications for all of the components involved in the counter register structure, we may propose the agent ImpCR defined by

$$\text{ImpCR}(A,B,c) = (\text{ImpCLC}(B) \times \text{ImpCLK}(c) \times \text{ImpT}(A,c) \times \text{ImpT}(B,c))\backslash\text{ComCR}(c) \tag{12}$$

as the low-level specification of the counter register behaviour.

Proposition 3. *For the current binary content AB of the register and the clock signal $c \in \{0,1\}$, the previous agents SpecCR(A,B,c) and ImpCR(A,B,c) are bisimulation equivalent.*

Proof. Because the bisimulation relation is a congruence, the target result is immediate, based on the appropriate bisimilarities already proved in previous Proposition 1. and Proposition 2. ■
According to (8) the expressions of the internal counter register components are
SpecCLC$(B) = \sum_{E,x\in\{0,1\}}(\alpha_E\beta_x\overline{TA}_p\overline{TB}_E : \text{SpecCLC}(B))$ with $p = E$ AND (x NXOR B) and
SpecCLK$(c) = CLK_c\overline{CLKA}_c\overline{CLKB}_c : \text{SpecCLK}(c)$.
Besides, following the previous expressions (7) we have
SpecT$(A,c) = \sum_{p\in\{0,1\}}(\text{CLKA}_c TA_p\overline{\delta}_A^{(1)} : \text{SpecStepT}(A,T,c))$ and
SpecT$(B,c) = \sum_{E\in\{0,1\}}(\text{CLKB}_c TB_E\overline{\delta}_B^{(0)} : \text{SpecStepT}(B,T,c))$
Hence, following the definition relation (11), the expression of the counter register specification is

$$\text{SpecCR}(A,B,c) = \sum_{E,x\in\{0,1\}}(CLK_c\alpha_E\beta_x\overline{\delta}_A^{(1)}\overline{\delta}_B^{(0)} : \text{SpecStepCR}(E,x,A,B,c)) \tag{13}$$

This theoretical result corresponds to the automatic verification of the same counter register modelled with CWB-NC platform. Let consider the behaviour of the counter register enabled to operate one step from the current content AB=00 to the next possible combinations according to the input values of the circuit. In the left-hand part of Figure 3 we represent the CWB-NC automaton built for the agent SpecCRAB00s1. In the right-hand part of the same figure we have the CWB-NC answer TRUE for the automatic verification of the bisimulation equivalence between the agents SpecCRABs1 and ImpCRAB00s1.

```
cwb-nc> compile SpecCRAB00s1
Building automaton...
States: 6
Transitions: 10
Done building automaton.
0:      'CLK1.E0.x0.~A0.~B0'    {1}
        'CLK1.E0.x1.~A0.~B0'    {1}
        'CLK1.x0.E1.~A0.~B0'    {2}
        'CLK1.x1.E1.~A0.~B0'    {3}

1:      'CLK1.E0.x0.~A0.~B0'    {1}
        'CLK1.E0.x1.~A0.~B0'    {1}

2:      'CLK1.x0.E1.~A0.~B0'    {4}

3:      'CLK1.x1.E1.~A0.~B0'    {5}

4:      'CLK1.x0.E1.~A1.~B1'    {4}

5:      'CLK1.x1.E1.~A0.~B1'    {5}

Start States:  [0]
```

```
cwb-nc> eq SpecCRAB00s1 ImpCRAB00s1
Building automaton...
States: 12
Transitions: 20
Done building automaton.
Transforming automaton...
Done transforming automaton.
TRUE
```

Figure 3: CWB-NC verification for the counter register

4 Conclusions

From the theoretical point of view, this paper uses the algebraic theory of processes as a formal method applied to model the behaviour of hardware components. This approach is also achieved in [1] for modelling the general

circuits behaviour in CCS or in [6] for modelling the flip-flops behaviour in VHDL. Beyond these natural models, in this paper we had to adjust a pure algebraic language like SCCS to model a specific type of hardware component, namely the counter register.
From the practical point of view, a counter register is used for counting the number of occurrences of a certain event. Based on an algebraic approach, this paper results refer to a counter register for one step operation, meaning the possibility of counting one occurrence of the event. It follows that our work will progress in the direction of modelling the intercommunicating one step counter registers.

References

[1] G. Clark, G. Taylor ”The Verification of Asynchronous Circuits using CCS”, *ECS-LFCS-97-369*, 1997.

[2] O. Georgescu, ”Problem Solving with Different Models”, *Proc. of SEEFM05 2nd South-East European Workshop on Formal Methods*, Ohrid, Nov.2005.

[3] Milner R., ”Calculi for synchrony and asynchrony”, *TCS*, Vol.25, pp.267-310, 1983.

[4] R. Milner, *Communication and concurrency*, Prentice Hall, 1989.

[5] A. Vasilescu, ”Algebraic Model for the JK Flip-Flop Behaviour”, *Proc. of SEEFM05 2nd South-East European Workshop on Formal Methods*, Ohrid, Nov.2005.

[6] R.D. Wittig, ”OneChip: An FPGA Processor With Reconfigurable Logic”, *IEEE Symposium on FPGAs for Custom Computing Machines*, 1995

[7] *** The CWB-NC homepage on http://www.cs.sunysb.edu/~cwb.

Anca Vasilescu, Oana Georgescu
Transilvania University of Braşov
Department of Theoretical Computer Science
Address: Str. Iuliu Maniu 50, 500091 Braşov, România
E-mail: {vasilex,o.georgescu}@unitbv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 465-469

Motion and Color Cues for Hands Detection in Video Based Gesture Recognition

Radu Daniel Vatavu, Stefan-Gheorghe Pentiuc

Abstract: The paper addresses the problem of hands detection in video sequences as the first step for gesture recognition based applications. Gesture recognition emerges as a natural human computer interface technology and reliable hands detection algorithms are needed as preliminary processing steps. Discussions are conducted on several techniques including motion and color as cues for detecting hands in real time video. Background subtraction algorithms and skin color detectors are investigated. An approach of complexity O(n*log(n)) that interpolates standalone results using each cue is equally proposed, where n is the dimension of a given video frame.

1 Introduction

Gestures are perceived as a natural mean of interacting and conveying information [1] hence a gesture based interface would prove ideal for the human computer interaction [2, 3]. Even more, video based gesture recognition is non intrusive, does not require the user to wear additional equipments or devices and gives a comfortable feeling of naturalness.

Hands detection in video sequences can be looked at as the first step in a video based gesture recognition application. In order for the gesture recognition component to be successful, prior stable and reliable hands detection algorithms are needed. The paper gives an overview of the existing color segmentation and motion detection techniques with application to detection of hands in video sequences. Particular approaches are discussed such as skin color modeling or background subtraction. A solution that considers both motion and skin color as cues for hands detection is proposed and the performances of the algorithm are discussed. The complexity of the approach is O(n*log(n)) where n is the dimension of the processed video frame, allowing for real time hands detection at a resolution of 320x240 and 25fps.

2 Color based segmentation

Skin color detection has proven to be an important intermediate step for face and hands tracking algorithms [4..11]. It has been observed on large image datasets [12..16] that skin color clusters under different limits in several color spaces. Based on this comfortable property, a few approaches have been proposed: histograms of skin probability at different resolutions [12]; single or mixtures of Gaussians modelling [12, 14, 15]; elliptical [17] or various curved and linear polygonal segments for skin cluster delimitation. Different properties of the existing color spaces have been investigated [13, 15, 16] and new color spaces were specially designed [16].

A common conclusion is that skin color indeed clusters under known limits in several color spaces. A simple and low cost procedure is to filter the current video frame using simple static thresholds in a given color space (for example we can have the interval $[h_{low}, h_{high}]$ for the hue component in the HSV color space)

$$p \text{ is skin} \iff hue(p) \in [h_{low}, h_{high}] \wedge saturation(p) \in [s_{low}, s_{high}] \quad (1)$$

where p is the current pixel submitted to classification and $[h_{low}, h_{high}]$ and $[s_{low}, s_{high}]$ are the low and high thresholds for the hue and saturation components.
The results of such an approach are presented in Figure 1. The technique is very fast and outputs all the true positives, however, the choice of fixed thresholds leads to disturbing segmentation results that may affect higher logic levels: over segmentation (false detections are very likely to appear if the thresholding interval is set too large) or under segmentation (false rejections in the case of a too tight interval). These effects are caused by the skin cluster position that may change in time due to various conditions: illumination changes, video noise, user skin color, etc.

Figure 1: Skin color detection result (Raw video frame, HSV conversion of the raw video frame, Skin color filtering result)

3 Motion detection and background subtraction techniques

The goal of background subtraction techniques [18, 19, 20, 21] is to detect all foreground objects by eliminating the background, considering a given video frame sequence captured from a static camera. The native idea that lies behind this approach considers a simple filtering of the current video frame by comparing it with a background distribution model, as follows:

$$|VF_i - B_i| \geq Threshold \tag{2}$$

where VF_i and B_i are the video frame and the background model at time i.

The problem consists in estimating of a background model to be used as a reference for all further comparisons. The background model must be updated with every new processed video frame in order to assure reliable results and to compensate for:
- changes in illumination (whether they are gradual such as the sunlight over the day or sudden such as when switching the indoor lights on/off)
- changes in motion (for example small changes introduced by camera movements that may oscillate and thus generating false motions or uninteresting motions that may be generated by real objects in the field view, others than the one the system is tracking Ű for example, video surveillance cameras may be fooled in triggering motion event detections on trees branch movements)
- actual background changes (the actual background may change in time, for example consider a camera monitoring the user hands working at a desktop from a top view. Objects may come in and out of the scene such as paper sheets, books, etc for different periods of time)

Basic motion detection techniques consider simple differencing between consecutive frames, as follows:

$$|VF_i - VF_{i-1}| \geq Threshold \tag{3}$$

In this simple case, the background model is represented by the previous video frame and thus brings a few disadvantages such as high dependency on the thresholdŠs value, poor detection of high speed objects, and dependency on the video frame rate.

A more reliable approach considers for the background model the average of the previous N video frames:

$$B_i = \frac{1}{N} \sum_{j=i-N}^{i-1} VF_j \tag{4}$$

or, for the purpose of saving memory resources as the running average:

$$B_i = \alpha \cdot VF_{i-1} + (1-\alpha) \cdot B_{i-1} \tag{5}$$

where α is the learning rate (for example between 0.02 and 0.1).

The classification results (foreground / background) may also be used for future background models. If a pixel

is classified as foreground by the current comparison, this information is used for the next background computation:

$$B_i(x,y) = \begin{cases} \alpha \cdot VF_{i-1}(x,y) + (1-\alpha) \cdot B_{i-1}(x,y) & \text{if } VF_{i-1}(x,y) \text{ is background} \\ B_{i-1}(x,y) & \text{otherwise} \end{cases} \quad (6)$$

Simple Gaussians or mixture of Gaussians distributions [21] have also been used for constructing the background models. For example, the running average technique may be updated to: B_i = (μ_i, σ_i) where $\mu_i = \alpha \cdot VF_{i-1}$ + (1 - α) $\cdot$ μ_{i-1} and $\sigma_i^2 = \alpha \cdot (VF_{i-1} - \mu_{i-1})^2 + (1-\alpha) \cdot \sigma_{i-1}^2$. The threshold value may be chosen as a multiple of σ.

The results of the running average technique are presented in Figure 2, below.

Figure 2: Background subtraction result (blue color indicates the background model)

4 Combining color and motion for hands detection

Considering the previous approaches that take into account skin color or motion detection, one can easily observe that the standalone results are not perfect. That is, false results are triggered by the skin color detector due to objects in the real world that present the same color as skin (for example, parts of the userŠs keyboard may be falsely detected as skin). Also, motions may be generated by other sources than the objects of interest (the userŠs hands), for example: chair motion, keyboard or mouse movements, etc. However, linear interpolation the results outputted by the two approaches can improve the final result:

$$HD_i = \lambda \cdot SKIN_i + (1-\lambda) \cdot MOTION_i \quad (7)$$

where λ is a constant that privileges color over motion (a value of 0.8 has been chosen). The final image is submitted to a blob identification module and the biggest blobs are identified as the userŠs hands. The results are presented in Figure 3 below.

Figure 3: Combining color and motion cues for hand detection (Skin color detection result, Background subtraction, Hands detection)

The proposed algorithm is given below:

*) Let VF_i be the current video frame; α =0.05 the background model learning rate; λ =0.8 the interpolation constant for combining the color and motion cues

*) Compute the color segmentation result $SKIN_i = VF_i | hue, sat \in [h_{low}, h_{high}] \times [s_{low}, s_{high}]$

*) Compute the motion detection result and update the background model
$MOTION_i = |VF_i - B_i| \geq Threshold$
$B_i = \alpha \cdot VF_{i-1} + (1-\alpha) \cdot B_{i-1}$

*) Compute the final result and apply blob identification
$HD_i = \lambda \cdot SKIN_i + (1-\lambda) \cdot MOTION_i$ *) Segment as hands the two biggest blobs identified

The final algorithm complexity is as follows:
1. Skin color detection (RGB to HSV conversion + HS filtering) = $O(n)$
2. Background subtraction (running average) = $O(n)$
3. Results interpolation $O(n)$
4. Blob connected components identification $O(n \cdot log(n))$
Total complexity is $O(n \cdot log(n))$ where *n* is the dimension of the processed video frame.

Considering the processing resolution of 320x240, the approach allows for real time hands detection at 25fps (n = 320 x 240 x 3 = 230400, n*log(n) = 230400 * 17.81 which is approx. 4000000 operations needed for processing a single video frame; 25 x 4000000 = 100000000 total operations needed for processing 25 fps, easily achieved by today's microprocessors).

5 Summary and Conclusions

An approach for detecting hands in real time video is presented by considering an interpolation of color and motion cues. Skin color models are discussed as well as background subtraction algorithms. The final technique consists in interpolating the standalone results from the two approaches by privileging color in a ratio of 8:2. The total complexity of the approach is $O(n \cdot log(n))$ where n is the dimension of a given video frame. The hands detection will serve as the preliminary process for a gesture recognition based human computer interaction system.

References

[1] Melinda M. Cerney, Judy M. Vance, "Gesture Recognition in Virtual Environments: A Review and Framework for Future Development," *Human Computer Interaction Technical Report ISU-HCI-2005-01*, 28 March 2005.

[2] Matthew Turk, "Gesture Recognition," *Chapter 10*, available at chapter 10, http://ilab.cs.ucsb.edu/projects/turk/ TurkVEChapter.pdf.

[3] Axel Mulder, "Hand Gestures for HCI," *Hand Centered Studies of Human Movement Project, Technical Report 96-1*, School of Kinesiology, Simon Fraser University, February 1996.

[4] Elli Angelopoulou, "Understanding the color of human skin," *Proc. SPIE Conf. on Human Vision and Electronic Imaging*, VI 2001, SPIE Press, pp. 243-251.

[5] Vladimir Vezhnevets, Vassili Sazonov, Alla Andreeva, "A survey on pixel based skin color detection techniques," *Proc. Graphicon* , Moscow Russia, pp. 85-92, 2003.

[6] J. Ruiz-Del-Solar, R. Verschae, "Robust skin segmentation using neighborhood information," *Int. Conf. on Image Processing*, pp. 207-210, 2004.

[7] Sanjay Kr. Singh, D. S. Chauhan, Mayank Vatsa, Richa Singh, "A robust skin color based face detection algorithm," *Tamakang Journal of Science and Engineering*, vol. 6, no. 4 (2003), pp. 227-234.

[8] K. Schwerdt, J. L. Crowley, "Robust face tracking using color," *4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Grenoble France, 2004.

[9] Shinjiro Kawato, Jun Ohya, "Automatic skin color distribution extraction for face detection and tracking," *Proc. 5th Int. Conf. on Signal Processing*, Beijing China, pp. 1415-1418, 2000.

[10] Prem Kuchi, Prasad Gabbur, P. Subbanna Bhat, Sumam David S, "Human Face Detection and Tracking using Skin Color Modelling and Connected Component Operators," *IETE Journal of Research*, vol. 38, pp. 289-293, 2002.

[11] S. Marcel, S. Bengio, "Improving face verification using skin color information," *Proc. 16th Int. Conf. on Pattern Recognition*, vol. 2, pp. 11-15, IEEE Computer Society Press, 2002.

[12] Michael J. Jones, James M. Rehg, "Statistical color models with application to skin detection," *TR 98/11*, Cambridge Research Laboratory, 1998.

[13] Tiberio S. Caetano, Silvia D. Olabarriaga, Dante A. C. Barone, "Do mixture models in chromaticity space improve skin detection?" *Pattern Recognition*, 36 (2003), pp. 3019-3021.

[14] T. S. Caetano, D. A. C. Barone, "A probabilistic model for human skin color," *Int. Conf. on Image Analysis and Processing*, pp. 279-283, 2001.

[15] Tiberio S. Caetano, Silvia D. Olabarriaga, Dante A. C. Barone, "Performance Evaluation of Single and Multiple-Gaussian Models for Skin Color Modelling," *Proc. 15th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 275-282, 2002.

[16] Giovani Gomez, "On selecting color components for skin detection," *Proc. Int. Conf. on Pattern Recognition*, 2002.

[17] Jae Y. Lee, Suk I. Yoo, "An elliptical boundary model for skin color detection," *Int. Conf. on Imaging Science, Systems and Technology*, Las Vegas USA, 2002.

[18] B.P.L. Lo, S.A. Velastin, "Automatic congestion detection system for underground platforms," *Proc. of 2001 Int. Symp. on Intell. Multimedia, Video and Speech Processing*, pp. 158-161, 2000.

[19] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russel, "Towards Robust Automatic Traffic Scene Analysis in Real-Time," *Proceedings of Int. Conference on Pattern Recognition*, pp. 126Ű131, 1994.

[20] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, "Pfinder: Real-time Tracking of the Human Body," *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. 19, no. 7, pp. 780-785, 1997.

[21] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *Proc. of CVPR.*, pp. 246-252, 1999.

Radu Daniel Vatavu, Stefan-Gheorghe Pentiuc
"Stefan cel Mare" University of Suceava
Department of Computer Science
Address: 13, Universitatii St, RO-720229 Suceava
E-mail: raduvro@yahoo.com, pentiuc@eed.usv.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 470-473

Data Integrity and Integrity Constraints in Databases

Mădălina Văleanu, Grigor Moldovan

Abstract: An important component of any data model, particularly a relational one, is the one with which one can specify the conditions to be fulfilled by data in a database. These conditions are called integrity constraints (restrictions). This paper presents the notion of statistical integrity and some integrity indicators with the purpose of generalizing data integrity as a concept. For the processing outcomes to be relevant in a certain application, it is required to use correct data. Data quality of being correct needs definition. A section of this paper is dedicated to a synthesis of the already famous definitions from the reference literature in relation with data integrity.

Keywords: data integrity, integrity constraints, statistical integrity

1 Introduction

The importance of integrity as a security objective has been increasingly recognized in recent years. System integrity is a more general term that is additionally concerned with integrity of the processing elements such as hardware, system and application software.
To have relevance in the data processing of a certain application, the data supplied should be *correct/true.* Data quality of being correct requires more specifications. Generally, data are facts collected from the real world based on observations and measurements. In some cases, hence, data are the results of measurements that implicitly contain some measurement errors ranging in an acceptable interval. In this case, data are presented by approximate numerical values. For instance, an individual's blood pressure is measured by means of a device and is expressed in whole numbers at a certain time moment that can yield other values at the times.
However, the notion of data integrity itself is not precisely and rigorously enough defined. There is no consensus in so far the significance of the integrity concept is concerned. References are rich in the numerous attempts to define data integrity [2], [3].

2 Data Integrity definitions

For the last years, data integrity has been considered and appreciated as very valuable. Unfortunately, no consensus exists related to the meaning to be given to this term. Five data integrity definitions will be shown below, their selection being made according to the author's appreciation according to significance and ranking.
In the literature, distinction is made between data integrity and system integrity. Data integrity deals with data themselves, i.e. bites and bits stored in the system. System integrity is a more general term that additionally refers to the integrity of the processing elements, such as hardware, system and application program. We shall restrict our comments to the definitions of the data integrity. In most cases, we assume that hardware and software do not exhibit disorders.
For data integrity definitions we refer to the following aspects:

a. The most general definition is the one that mentions the data quality, given by Courtney and Ware [4]. This is based upon the idea that we expect data quality: data exhibit integrity to the extent to which their quality satisfies or exceeds the quality requirements expected by the user. This definition is the only that contains viability conditions. For example, the temporal quality can deteriorate when data is updated regularly.

b. Other definitions rely on the capacity to modify data, thus being that of Sandhu and Jajodia [5] that defines integrity as a measure of protection *to incorrect data modifications*.

c. Another definition narrows the protection to modification even more, as *unauthorized data modifications are considered*. This last definition was extracted from the data security criteria and they are called *definitions on data modifications*.

d. The fourth definition discusses Biba's concept that integrity is *a one-direction information flow* in a network [6]. Biba's definition is still more restrictive than the first three as it gives a very specific meaning to the unauthorized data modification. This is called the *information flow definition*.

e. The fifth and the narrowest definition originate in the network field and *require data not to be modified o*r at least any modification brought to be detected. This definition can be noticed due to its comprehensive character and its position at the end of this list.

3 Possible ranking of the five definitions mentioned

By comparing the five data integrity definitions, we can arrange them in an increasing order so:

- Courtney-Ware's data quality definition [4] includes viability (something good will occur) and safety (i.e. there will not happen something bad), as requirements;
- Sandhu and Jajodia's definition [5] on the improper modification of data limiting to safety requirements,
- The definition including the unauthorized modification of data is popular in more recent criteria but it is limited to access control,
- Biba's definition [6] deals with access control to ensure a one-way low of information in a network (that is not different from the network based definitions of confidentiality) [7][8];
- Definition of network security requires data not to be modified (or at least that modifications are detectable) and this is the most restrictive of them.

Function of the database type, we may need a „perfect integrity" or we may admit a certain tolerance for the criteria at the basis of integrity. Thus, for financial and banking, for ticket reservation databases, the integrity to be ensured should be very restrictive to reach the perfect model as nearly as possible. In other types of databases the distance to the perfect model can be larger. Thus, data tolerance can be defined as:
c*_data_, where $c \in [-a, a], a \geq 1$.
For c we can perform the transformation: $c = f$*[0,1].
It would be interesting to see whether a similar ranking of definitions for data confidentiality could be made.

4 A generalization of the concept of data integrity

The concept of data has evolved a lot. At the beginning, data meant a numerical magnitude and then it started enlarging in concept. The evolution continued when by data one meant a group of data constituted in various types of structures, as for example: table, file or even database. Now, we generically call *data* any of the structures mentioned here.
Data integrity refers to elementary data or to data forming a certain structure. The integrity of a data of a given structure is determined by the integrity of all elementary data forming the respective structure.
Data integrity is a feature measuring data semantic aspects, such as:

- correct data
- precise data
- pure date
- exact data

In this context, a certain data *quality* achieved by constraints is highlighted.
How liberal could a computer system controlling the data integrity be? One should always interpose *some* constraints; question is how many? It is obvious that a database application is operationally efficient if an optimum balance between data liberal features and integrity constraint check complexity level can be established.
We can find out that using the definitions above data integrity can be assessed with integrity indicators. We could have general integrity indicators as well as specific ones. These indicators can be analytical or static in nature. [1]

Examples. Be the set of numerical data with approximate values $\{x_1, x_2, \ldots, x_n\}$. Note the exact values with $\{y_1, y_2, \ldots, y_n\}$. For the measurement of data integrity we can consider:
$A = \sum_{i=1}^{n} |x_i - y_i|, B = \max_i |x_i - y_i|$, $C = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

5 Statistical integrity

The concept of statistical integrity has been long used in a different approach than the present one. Recently, this data property was the topic of the research relative to data containing *confidential information*, as for example, on the health state of a person, on the financial state of a company or data relating to a business. A special case is represented by microdata that are statistical series of recordings, tuples respectively; in each recording there is information on an individual unit (entity), such as a person, a firm, an institution etc. Microdata can be represented as a relational database, whose tuple is an individual unit. To ensure confidentiality for some data part, if not all, of the values of the relationship scheme are values transformed by means of a given function. In this way, parts of the values of the attributes of a relational database are called *marked*. For medical databases that in general are statistical in nature, the guarantee of anonymity has a very large importance in some cases. Most often, this guarantee is regulated by the legislation in the field.
Statistical data are reached, in general, from more or less accurate measurements, i.e. a certain error tolerance will always be present.
If x_1, x_2,...,x_n is a series of statistical data [9] then it represents only approximate values of some concrete magnitudes. Generally, these data measuring a certain characteristic (for example, medical), is replaced by the average of these values $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$. Probabilistically, we speak of a random variable x with distribution $\begin{pmatrix} x_1 x_2 \ldots x_n \\ p_1 p_2 \cdots p_n \end{pmatrix}$, $\sum_{i=1}^{n} p_i = 1$. Then, the average value of these random variables is $\overline{x} = \sum_{i=1}^{n} x_i p_i$.
We immediately find that the deviations average is null, hence this average does not describe in different manners two variables with the same average. In fact,
$x - \overline{x} : \begin{pmatrix} x_1 - \overline{x} & x_2 - \overline{x} & \ldots & x_n - \overline{x} \\ p_1 & p_2 & \ldots & p_n \end{pmatrix}$, and
$\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x}) = \sum_{i=1}^{n} (x_i - \overline{x}) p_i = \sum_{i=1}^{n} x_i p_i - \overline{x} \sum_{i=1}^{n} p_i = \overline{x} - \overline{x} = 0$

This demonstration represents the logical reasoning of the doubts on statistical average values (as the statement that, in average, something is good, hides blatant inequalities).[1]
The deviations average from the average is not conclusive so that it is necessary to adapt a *differentiation indicator and* to define the average of the deviations average square as:
$\sigma_x^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2$

With a simple calculation we find
$\sigma_x^2 = \overline{x^2} - \overline{x}^2$, where $\overline{x^2} = \sum_{i=1}^{n} {x_i}^2 p_i$.
We note then $\sigma_x = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2} \equiv D\ (x)$ that represents the *square average deviation or the standard deviation or standard dispersion.* $\sigma_x$with a lowvalue means that we can „trust" the average calculated. Dispersion represents the second order moment. A more refined indicator would be some order indicator *r, r≥2*, that is:
$D^r(x) = \sum_{i=1}^{n} (x_i - \overline{x})^r p_i$. (for discrete random variables),

respectively

$D^r(x) = \int \sum (x_i - \overline{x})^r \rho(x) d(x)$ (continuously).

Often, statistical data originate in the measurements of some characteristics, i.e. in the findings of some variables magnitudes. The manner in which decision is made in so far the highlighting of the random change in a variable, due to an *error* or a systematic modification occurred is extremely important. There are criteria that can be used to establish whether some measurement deviations can be taken as errors or systematic modifications.

If e_i is the measurement error for a variable, then the variable average magnitude should be zero or near zero. Another criterion could be that of dispersion. The dispersion indicator should not differ too much from one set of measurements to another. Finally, the laws of random variations should also be considered. For example, in statistics, one applies the normal distribution law (Gauss-Laplace) that defines a bell-shaped curve for a random variable:

$$f(x) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(x-\bar{x})}{2\sigma_x^2}}$$

To admit an error, it must be defined by the following:

- to be by chance, each error magnitude being characterized by a probability coefficient;
- to have an average null value;
- to have a constant variation (dispersion) for any set of values of the random variable;
- to have a normal error distribution.

6 Summary and Conclusions

A database is taken as a "good database" only when the data stored in it are correct. It is for this reason that any DBMS should prevent the insertion of incorrect data in the database. Consistency constraints also often called integrity constraints or integrity rules in the references.

References

[1] M. Moldovan, "Problema integritatii în baze de date distribuite" , PhD thesis, "Babes-Bolyai" University, Cluj-Napoca, 2004.

[2] R. S. Sandhu, "Lattice-Based Access Control Models," *IEEE Computer*, vol 26, nr. 11, 1993.

[3] R. Ramakrishnan, "Database Management System" *Mc Garw-Hill*, 1998.

[4] R. Courtney , "Some Informal Comments about Integrity and the Integrity Workshop," *Proceedings of the Invitational Workshop on Data Integrity*, 1989.

[5] R.S. Sandhu, S.I. Jajodia, "Integrity Mechanisms in Database Management Systems," *Proceedings 13th National Computer Security Conference*, 1990.

[6] K.J. Biba, *Integrity Considerations for Secure Computer Systems*, Mitre Corporation, Belford, Massachusets, 1977.

[7] D.E. Denning, "A Lattice Model of Secure Information Flow," *Communications of ACM* , Vol. 19(5), 1967.

[8] S. Tigan, A. Achimas, T. Drugan, "nformatică si statistică aplicate în medicină," *Ed. Srima*, 2001.

[9] P. Blaga, "Statistică matematică," lito. Univ. Babes-Bolyai, Cluj-Napoca, 2000

Mădălina Văleanu
University of Medicine and Pharmacy "Iuliu-Haţieganu"
Departament of Medical Informatics and Biostatistics
Cluj-Napoca
E-mail: mvaleanu@umfcluj.ro

Grigor Moldovan
Babes-Bolyai University
Department of Computer Systems
Cluj-Napoca
E-mail: moldovan@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 474-479

Computational Inteligence-based Model for Component Composition Analysis

Andreea Vescan, Laura Dioşan

Abstract: Component-based systems are built by assembling components developed independently of the systems. A challenge in component-based software development is how to assemble components effectively and efficiently. In this paper we use genetic algorithms for component composition analysis. Starting from an automaton-based model and using integration properties we develop a genetic algorithm to analyze the component composition process. System features and genetic representation are given. Some experiments are performed.

Keywords: Component-based System, Software Component, Genetic Algorithm.

1 Introduction

Specification of software components is one of the most important research challenges in component-based software engineering. It represents the first step toward true component reuse as the component specification gives all necessary information to the component user on how/why the component can be (re)used.
Grady Booch believes, that "a reusable software component is a logically cohesive, loosely coupled module that denotes a single abstraction" [1]. Szyperski describes a software component as "a unit of composition with contractually specified interfaces and explicit context dependencies only" that "can be deployed independently and is subject to composition by third parties." He also argues that components have no state and are binary deployable [9]. According to MetaGroup (OpenDoc), "Software components are defined as prefabricated, pre-tested, self-contained, reusable software modules - bundles of data and procedures - that perform specific functions." Intuitively, a component is designed to be composed [4], that is, it is a basic unit for composition. Still, a component is not everything; its structure and behaviour should follow specific rules. Thus, "a software component is a software element that conforms to a component model and can be independently deployed and composed without modification according to a composition standard."[3]
The component in its simplest form contains code that can be executed on some platform and an interface that provides the access to the component. A component is considered to be a black box, i.e., its internals inaccessible to the component user. Hence, interfaces are the only access points to the component and the specification of the component comes down to the specification of the component interfaces. Specification of the component interfaces [5] in the current component-based systems is done only on the syntactical level. On this level, the component specification consists of specification of provided and required interfaces. Provided interfaces are the one that contain operations that a component provides to other components or to the component user, while required interfaces are the one that contain operations used by the component. Each interface consists of a number of operations which might have input, output or input/output parameters.
In this paper we use the component model from [7] and [2]. The paper proposes a model of a component-based software system, which uses Genetic Algorithms (GAs), enabling component composition analysis. The mapping steps from an automaton-based model to a genetic algorithm-based model are given in the following sections.

2 Proposed model

The model proposed in this paper is obtained from an already developed automaton-based model. We use the some features and properties from the given model but encoded into a genetic algorithm representation.

2.1 A finite automata-based model

In [7] a model of a component-based software system is proposed, which uses a finite automaton-based method, enabling compositional reach ability analysis. The following checks were performed:

- the system is consistent: starting from a given input, all components can be added to the model and the execution eventually terminates;

- there are no potential deadlocks in the model.

The model is used forward in [2] to construct all the component-based software systems as finite automaton-based models. The resulting models are syntactically correct. By syntactically correct model we mean no semantic involvement in the models, but just the way to connect the components together, the mechanical process of "wiring" components together (component integration). The algorithm checks model's consistency during its construction from a given set of components. The result systems are checked against some properties: if the execution of the component system is terminated; if the system behaves properly (if there are some lost data) and a component is not allowed to receive the value for an inport from more than one component - one provider/inport.
Consider the component system $CS = \{C_1, C_2, ..., C_n\}$, in which every component C_k is specified as:

$$C_k = (compID_k, inports_k, outports_k, functs_k),\ where$$

- $compID_k$ is the component identifier, unique;
- $inports_k$ the set of input ports and $outports_k$ - the set of output ports;
- $functs_k$ the set of tasks the C_k component performs.

Example. Consider the following general set of components:
$C_1 = (C1, \varnothing, \{d_1, d_2\}, \{read\})$;
$C_2 = (C2, \{d_1, d_3\}, \{d_5, d_6\}, \{task_1, task_2, task_3\})$;
$C_3 = (C3, \{d_2\}, \{d_3, d_7\}, \{task_4\})$;
$C_4 = (C4, \{d_5, d_7\}, \{d_8\}, \{task_5\})$;
$C_5 = (C5, \{d_6, d_8\}, \varnothing, \{write\})$;
$C_6 = (C6, \{d_1, d_3\}, \{d_4, d_5, d_6\}, \{task_1, task_2, task_3\})$;

The first component that is used from the component system is a source component [1]. A component-based system is found when the last component added to solution is a destination [2] component.
The solutions from Figure 1 starts with the evaluation of the first component and contains a different number of components involved into the computation. The solution from figure 1.a) shows two different cases that are not wanted for the final solution: lost data and having more than one provider/inport.

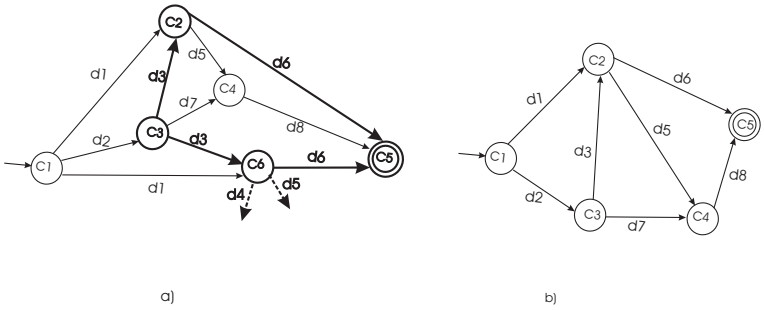


Figure 1: The finite automaton a) with lost data $d4$ and $d5$ and component $C5$ receiving data for inport $d6$ from component $C2$ and $C6$; b) corresponding to the final consistent solution.

The component $C6$ outport contains data $d4$ and $d5$ which are lost (no other component from the system is using it). Also, the solution is not included in the solution set with one provider/inport because there are two transitions to the same component $C5$ with the label $d6$. The input $d6$ of the $C5$ component must have only one provider on an execution. The "reverse" propagation of data (the output data of a component is propagated to two or more components) is allowed. Component $C3$ distributes the data $d3$ to $C2$ and $C6$ components.
In figure 1.b) the final solution is presented: the solution is consistent, no lost data and each inport for each component has just one provider.
The final consistent solution is $A = (Q, \Sigma, \delta, q_0, F)$, where: $Q = \{C1, C3, C2, C4, C6, C5\}$, $\Sigma = \{d1, d2, d3, d7, d5, d6, d8, d4\}$, $\delta = \{(C1, d2) \rightarrow C3, (C1, d1) \rightarrow C2, (C3, d3) \rightarrow C2, (C2, d5) \rightarrow C4, (C3, d7) \rightarrow C4, (C2, d6) \rightarrow C5, (C4, d8) \rightarrow C5\}$, $q_0 = \{C1\}$ and $F = \{C5\}$.

[1] A source component, i.e. a component without inports, is a component that generates data provided as outports in order to be processed by other components.

[2] A destination component, i.e. a component without outports, is a component that receives data from the system as its inports and usually displays it, but it doesn't produce any output.

2.2 A computational inteligence-based model

Genetic Algorithms [6] are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetic operations. The basic concepts of GAs are inspired by the natural process from real-life. They explore randomly the search space of solutions for a problem using three important genetic operators: replication, mutation and crossover. The search space is represented like a population of individuals, each of them can being a potential solution for the problem. At the begin, individuals are randomly initialize and during GAs iterations, using genetic operations, they are perturbed (oriented) to the optimal solutions of problem based on their quality (or fitness function).

Three main steps must be passed for solving a particular problem with GAs:

- the coding of the possible solutions for the problem;
- the evaluation process performed for each potential solution (the solution quality);
- the definition of the genetic operators that straighten the search process to the optimal one.

Individual representation. The proposed model uses a population of individuals, each individual being represented by a genes string. The length of a string (or the chromosome length) is equal to the components number that is used by the system. Each gene corresponds to a certain system component and it contains a number of ales equal to the number of outputs for corresponding component. Each ale represents the component index that uses the respective outport for the current component.

For instance, if we have a component $C4$ with 2 inputs and 4 outputs (outports) o_1, o_2, o_3, o_4 than the associated gene to this component can be: $(3, 1, 7, 5)$. It means that the third component of the system uses like input the first output of $C4$, the first system component can use the second output of $C4$, component seven can use the third output of $C4$ and, finally, the fifth component of the system uses o_4. But it is possible that not all elements from the initial specification will be included into the final system (final solution). Some data can be lost when we construct the optimal solution. Because of this reason a chromosome has attached a binary string that specify what components are keep in the final solution and what are not (1 if component take part to the final system and 0 otherwise).

The first operation that it is need to perform in a GA is the population initialization and we describe it in few words. Each ale of a gene is randomly initialized with a component index, but that it differ from the current component (or current gene index) - because we suppose that there are not "curls" (loops) in system (one of a component outports must be used by the other component and not by the parent component).

For instance, a chromosome associated with example presented in Section 2.1 can have the next form:

$$\underbrace{[1(2,3)]}_{C_1}\underbrace{[1(4,5)]}_{C_2}\underbrace{[1(2,4)]}_{C_3}\underbrace{[1(5)]}_{C_4}\underbrace{[1()]}_{C_5}\underbrace{[1(4,2,1)]}_{C_6}$$

where each gene is represented like $[presence\_bit(list\ of\ neighbors\ components)]$. Therefore, in this chromosome all components are used (all components have associated value 1) and for the first component the first output is associated with the second component and the second output is associated with the third component; for the second component its outputs are associated with fourth and fifth components, for outputs of the third component are associated second and fourth components, and so on.

Fitness measure. The search process in the problem solution space is based on the quality of individuals. The fitness measure plays the role to guide the search process when two individuals are selected for some genetic transformations. Each gene of a chromosome must be evaluated for computing the associated fitness measure. During this evaluation, we analyze the matching of the current component with it neighbors an, also, the "utility" of these matching.

Concerning the matching analysis, for a particular gene (component) i and its values x, y, z (we suppose that the component i has three outports) we verify if:

- outport o_1 can be input for the x-th component of the system;
- outport o_2 can be input for the y-th component of the system;
- outport o_3 can be input for the z-th component of the system.

We verify these conditions and we increase the fitness function with a unit for each of them that are not respected (we penalize the chromosome for each violated condition).

To establish the "utility" of matching process we verify if the necessarily information is provided for all system inports. We increase the fitness with the difference between the number of inports for all system components and the number of "used" inports (we quantify the "un-satisfied" inports). Also we verify if there are more than one

provider/inport. If we find elements that not respect these conditions we penalize the current chromosome (that is equivalent to increase the fitness value with a unit for each violated condition).
For precedent chromosome example the fitness is equal to 5 because after evaluation:

- no information is provided to inports of six-th component (two violated conditions),
- all the outports of sixth component are lined with components that don't need that data (component 4, 2 and 1 don't need $d4$, $d5$, $d6$ data, respectively - three violated conditions).

Genetic operators. Proposed model uses only crossover and mutation operations during the "adaption" process.
Crossover operation supposes to combine two chromosomes (called "parents") to obtain a new one that "brings" genetic information from both parents. Like in natural crossover, an offspring takes some parts of genetic material from his "mother" and other parts from his "father". Concerning crossover operator used in this model, it is one-cutting-point crossover. This type of crossover supposes to randomly choose a cutting-point in the two parent chromosomes (M and F) and "construct" the new individual (the offspring - O) using the genes from M place before cutting-point and the genes from F placed after the cutting-point. Note that exchanging an entire gene with other gene it is not possible to loose genetic information (all genetic operators must assurance the consistency and the validation of he obtained offsprings). For instance, we choose the position 2 as cutting-point for these two parent chromosomes:

$$P_1 = \underbrace{[1(2,3)]}_{C_1}\underbrace{[1(4,5)]}_{C_2}\underbrace{[1(2,4)]}_{C_3}\underbrace{[1(5)]}_{C_4}\underbrace{[1()]}_{C_5}\underbrace{[1(4,2,1)]}_{C_6} \quad P_2 = \underbrace{[0(5,3)]}_{C_1}\underbrace{[1(1,3)]}_{C_2}\underbrace{[0(6,1)]}_{C_3}\underbrace{[1(3)]}_{C_4}\underbrace{[1()]}_{C_5}\underbrace{[0(1,3,2)]}_{C_6}$$

The offspring obtained is

$$O = \underbrace{\overbrace{[1(2,3)]}^{P_1}\underbrace{[1(4,5)]}_{C_1}\underbrace{[1(4,5)]}_{C_2}}_{P_1}\underbrace{\overbrace{[0(6,1)]}^{P_2}\underbrace{[1(3)]}_{C_3}\underbrace{[1()]}_{C_4}\underbrace{[0(1,3,2)]}_{C_5}\underbrace{[0(1,3,2)]}_{C_6}}_{P_2}$$

Concerning *mutation*, our algorithm randomly chooses a component (or gene) i and for this gene it chooses an ale (one of component neighbors) and exchanges it with another one (randomly generated, but it is not the same with i). For precedent example we can select the second ale of the third gene for mutation. The mutated chromosome is:

$$\underbrace{[1(2,3)]}_{C_1}\underbrace{[1(4,5)]}_{C_2}\overbrace{\underbrace{[1(2,\overbrace{6}^{\text{mutated ale}}]}_{C_3}}^{\text{mutated gene}}\underbrace{[1(5)]}_{C_4}\underbrace{[1()]}_{C_5}\underbrace{[1(4,2,1)]}_{C_6}$$

Algorithm. The algorithm used for evolving a component composition system is described in this section. We use a generational evolutionary model as underlying mechanism for our implementation. The GA starts by creating a random population of individuals. Our algorithm uses another population (an auxiliary or intermediate population). The following steps are repeated until a given number of generations is reached: the best chromosome from current population is copied into intermediate population (elitism procedure). Then, until we fill the intermediate population we perform three operations: 1) two parents are selected using a binary tournament selection procedure; 2) the parents are considered for crossover, obtaining an offspring O; 3) we apply mutation to O and we obtain a new individual O' that is copied into the intermediate population.
After the auxiliary population is completed we copy all individuals (a generation) from auxiliary population into current population, starting a new generation.

2.3 Experiments

Example 1. Consider the following general set of components:
$C_1 = (C1, \varnothing, \{d_1, d_2\}, \{read\})$;
$C_2 = (C2, \{d_2, d_4, d_6\}, \{d_5\}, \{task_1, task_2, task_3\})$;
$C_3 = (C3, \varnothing, \{d_4\}, \{read\})$;
$C_4 = (C4, \{d_1\}, \{d_3, d_6\}, \{task_4\})$;
$C_5 = (C5, \{d_3, d_5\}, \varnothing, \{write\})$.
The solution is $A = (Q, \Sigma, \delta, q_0, F)$, where: $Q = \{C1, C3, C4, C2, C5\}$, Σ =$\{d1, d2, d4, d3, d6, d5\}$, δ=$\{(C1, d1) \rightarrow C4, (C1, d2) \rightarrow C2, (C3, d4) \rightarrow C2, (C4, d6) \rightarrow C2, (C2, d5) \rightarrow C5, (C4, d3) \rightarrow C5\}$, q_0=$\{C1\}$ and $F = \{C5\}$.

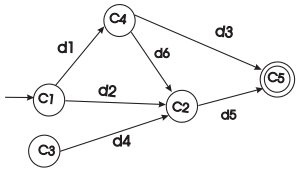


Figure 2: The finite automaton corresponding to the final consistent solution from example 1.

The best chromosome encodes the optimal configuration and is represented as the automaton from figure 2:

$$\underbrace{[1(4,2)]}_{c_1}\underbrace{[1(5)]}_{c_2}\underbrace{[1(2)]}_{c_3}\underbrace{[1(5,2)]}_{c_4}\underbrace{[1()]}_{c_5}$$

Example 2. Consider the following general set of components:
$C_1 = (C1, \varnothing, \{d_1, d_3\}, \{read\})$;
$C_2 = (C2, \{d_1, d_2, d_3\}, \{d_3\}, \{task_1, task_2, task_3\})$;
$C_3 = (C3, \varnothing, \{d_2\}, \{read\})$;
$C_4 = (C4, \{d_3\}, \{d_2, d_3\}, \{task_4\})$;
$C_5 = (C5, \{d_2, d_3\}, \varnothing, \{write\})$.
For the given set of components there are more solutions for the composition. Two possible solutions are represented in figure 3.

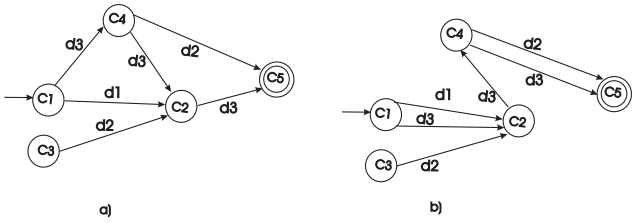


Figure 3: Two finite automata (from example 2) solutions obtained: a) $(C1, d3) \to C4$, $(C4, d2) \to C5$, $(C1, d1) \to C2$, $(C3, d2) \to C2$, $(C4, d3) \to C2$ and $(C2, d3) \to C5$; b) $(C1, d3) \to C2$, $(C1, d1) \to C2$, $(C3, d2) \to C2$, $(C4, d3) \to C5$, $(C4, d2) \to C5$ and $(C2, d3) \to C4$.

The automaton from figure 3.a) has the following chromosome representation

$$\underbrace{[1(2,4)]}_{C_1}\underbrace{[1(5)]}_{C_2}\underbrace{[1(2)]}_{C_3}\underbrace{[1(2,5)]}_{C_4}\underbrace{[1()]}_{C_5}$$

and from figure 3.b):

$$\underbrace{[1(2,2)]}_{C_1}\underbrace{[1(4)]}_{C_2}\underbrace{[1(2)]}_{C_3}\underbrace{[1(5,5)]}_{C_4}\underbrace{[1()]}_{C_5}.$$

The following chromosome contains a cycle: the $C4$ component expects data $d3$ from $C2$ component and the component $C2$ expects the output $d3$ data from the $C4$ component:

$$\underbrace{[1(2,5)]}_{C_1}\underbrace{[1(4)]}_{C_2}\underbrace{[1(5)]}_{C_3}\underbrace{[1(2,2)]}_{C_4}\underbrace{[1()]}_{C_5}.$$

3 Summary and Conclusions

In this paper we have introduced a new computational intelligence-based method for component composition analysis. Starting from a automaton-based model and using integration properties we develop a genetic algorithm to analyze the component composition process. Given a set of components to be integrated into a system we evolve

the solution (the component-based system(s)) using genetic algorithms. Some properties for the correct composition of components and component call dependences between components were encoded into the evaluation of a chromosome.

References

[1] Booch, Grady, *Software Component with ADA. 1st edition Software Component with ADA*, 1st edition. Benjamin-Cummings Publishing Co., Inc, 1987

[2] A. Fanea, S. Motogna, L. Dioşan, Automata-based Component System Building, *Studia Universitas "Babes-Bolyai", Seria Informatica*, L (2), 2006,(accepted).

[3] G. T. Heineman and W. T. Councill, editors, *Component-Based Software Engineering*, Addison Wesley, 2001.

[4] O. Nierstrasz and L. Dami, *Component-oriented software technology*, In O. Nierstrasz and D. Tsichritzis, editors, Object-Oriented Software Composition, pages 3-28. Prentice-Hall, 1995.

[5] J. Fredriksson, J. Hammarberg, J. Huselius, J. Håkansson, A. Karlsson, O. Larses, M. Lindgren, G. Mustapic, A. Möller, M. Nolin, T. Nolte, J. Norberg, D. Nyström, A. Tesanovic, M. Åkerholm Component Based Software Engineering for Embedded Systems A literature survey, *MRTC-Report* no. 102 ISSN 1404-3041 ISRN MDH-MRTC-102/2003-1-SE, 2003.

[6] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Boston, USA, 1989.

[7] B., Parv, S., Motogna, D. Petraşcu, Component System Checking Using Compositional Analysis, *Proceedings of the International Conference on Computers and Communications*, 2004, Baile Felix Spa-Oradea, Romania, pp. 325-329, 2004.

[8] Szyperski C., *Component Software, Beyond Object-Oriented Programming*, ACM Press, Addison-Wesley, NJ,1998.

Andreea Vescan, Laura Dioşan
Babeş-Bolyai University
Faculty of Mathematics and Computer Science
Computer Science Department
M. Kogalniceanu nr. 1, RO-400084 Cluj-Napoca, Romania
E-mail: {afanea, lauras}@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 480-485

Evolutionary Approach for Behaviour Component Composition

Andreea Vescan, Laura Dioşan

Abstract: Component-based software engineering (CBSE) is the emerging discipline of the development of software components and the development of systems incorporating such components. A challenge in component-based software development is how to assemble components effectively and efficiently. In this paper we present a new approach for behaviour component composition using Cartesian Genetic Programming. Some numerical experiments are performed.

Keywords: Component-based System, Software Component, Cartesian Genetic Programming.

1 Introduction

Building software with components promises more efficient and effective software reuse and higher productivity. The main advantage of component-based system development is the reuse of components when building applications. Instead of developing a new system from scratch, already existing components are assembled to give the required result.
There are two issues [2], [10] which need to be addressed where a software system is to be constructed from a collection of components:

- Component integration - the mechanical process of "wiring" components together. There has to be a way to connect the components together.
- (Behaviour) Component composition - we have to get the components to do what we want. We need to ensure that the assembled system does what is required [3], [4]. Component integration is taken one step further to ensure that assemblies can be used as components in larger assemblies. Component composition focuses on emergent assembly-level behaviour, making certain that the assembly will perform as desired and that it could be used as a building block in a larger system. The constituent components must not only plug together, they must play well together.

The matter of making pieces of software which will fit together has been the subject of considerable effort and systems and schemes exist which address these issues. These arrangements work by managing and controlling the interfaces between components.
The other problem is more subtle and difficult. We need to ensure that the assembled system does what is required. Just because the interfaces between two components permit them to be connected is no assurance that they will interact properly or even that making the connection makes any sense at all. The real solution to this problem is to reason about and check the behaviour of a system and the components from which it is built. Obviously, the definitive answer to the question, "if we put these components together like this, will it work?" can only be obtained by doing it and trying the completed system. But, if we are to avoid wasting time and effort, we want to know the answers before we build when the real system is not yet available. So we have to use something else and this is where models can help.
This paper proposes a model of a component-based software system, which uses Cartesian Genetic Programming (CGP), enabling behavioural system composition from a set of specified components. The mapping elements from a system involving components to CGP, the representation and the algorithm are given in the following sections.

2 Mapping Component-based Systems to Cartesian Genetic Programming

Cartesian Genetic Programming (CGP) [8] was introduce as an alternative methodology to standard Genetic Programming (GP), and secondly, to extend GP to non-Boolean problems and to show that it is a useful method for evolving programs with other data types [5], [6], [7]. CGP is Cartesian in the sense that the method considers a grid of nodes that are addressed in a Cartesian coordinate system.

Definition 1. A Cartesian program (CP) denoted P is defined as a set $\{G, n_i, n_o, n_n, F, n_f, n_r, n_c, l\}$ where: G represents the genotype and is itself a set of integers representing the indexed n_i program inputs; n_n represents the node input connections and functions; n_o represents the program output connections; the set F represents the n_f functions of the nodes; the number of nodes in a row and column are given by n_r, n_c respectively; program interconnectivity is defined by the levels back parameter l, which determines how many previous columns of cells may have their outputs connected to a node in the current column.

The genotype is a list of integers that encodes the connections and functions. It is a fixed length representation in which the number of nodes in the graph is bounded. However it uses a genotype-phenotype mapping that does not require all nodes to be connected to each other. This results in a bounded variable length phenotype. Each of the nodes represents a particular function and the number of inputs and outputs that each node has, is dictated by the arity of function.

The nodes take their inputs in a feed forward manner from either the output of a previous node or from one of the initial program inputs (terminals). The initial inputs are numbered from *0* to *n-1* where *n* is the number of initial inputs. The nodes in the genotype are then also numbered sequentially starting from *n* to *m+n-1* where *m* is the user-determined upper bound in the number of nodes. These numbers are used for referencing the outputs of the nodes and the initial inputs of the program. If the problem requires *k* outputs then these will be taken from the outputs of the last *k* nodes in the chain of nodes.

The general definition of a software component is given in [9] and highlights the basic characteristics of a component: an independent software module; provides functionality, but is not a complete system; can be accessed only through its interface; "black box" principle: that is, a component can be incorporated in a software system without regard to how it is implemented.

The definitions of a CP and of a component have given the idea that a system involving components might be generated using CGP, in which we have identified: the components are the nodes from the grid and the components execution order, the execution rules and the components interconnections are incorporated in the genotype.

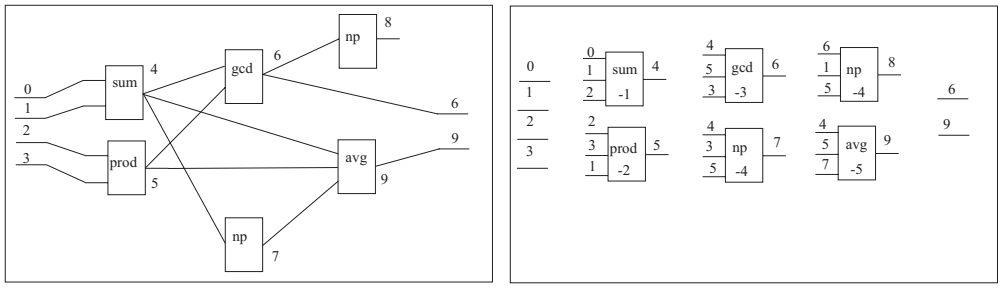


Figure 1: The mapping of a component-based system to a cartesian genetic program. a) The component-based system; b) The cartesian genotype.

In Figure 1 an example of mapping a component-based system to a cartesian genetic program is given. The system (chromosome) has four main inputs and two outputs. The components involved into the composition are: $sum(a,b) = a + b$ (index -1); $prod(a,b) = a * b$ (index -2); $gcd(a,b) = the\ greatest\ common\ divisor$ (index -3); $np(a) = next\ prime\ number\ greater\ than\ a$ (index -4) and $avg(a,b,c) = average\ of\ a,\ b\ and\ c$ (index -5). The system contains five distinct components and one component is used twice in the composition; each of these components need one or two or three input parameters and provide a single output. For example, the *np* component uses only one input from the input parameters, the *gcd* component uses two parameters and the *avg* component uses all the three input parameters. All this situations are better visualized into the component-based system representation. Another difference into the graphical representation of the component system and cartesian program is the dependences between the output of a component and the input of another component. The cartesian dependence is incorporated into the genotype rules.

3 Proposed Model

3.1 Representation

The CGP can provide the support that it is necessarily to evolve the structure of some component based systems. Such system can be viewed like a CGP chromosome. Each node (component) has a number of inputs and one or more functions that give the output(s) based on the inputs. In standard GP the evolved program only has one output (although the output could be a vector value); in CGP it is possible to have as many outputs as necessary. A component can have just one or many results.

Note that a node can have more inputs that it needs. For instance, if a node with two inputs must compute the greatest common divisor of two numbers, then it will use both parameters, but if a node with two inputs has associated a function that verifies if a number is a prime number or not, than only the first input will be used.

Another important remark concerns the chromosome parts that are the subject of adaptation process. During mutation it is possible to change the input(s) of a node, the function(s) associated to a node or the outputs of the system. The system inputs are pre-fixed.

The genotype (the list of integers) determines the connectivity and functionality of the nodes. These can be mutated over to create new directed graphs. When a population of genotypes is created or mutation is applied the genes must obey certain constraints in order for the genotype to represent a valid program [8].

Specific to genetic programming methods, in general, there are differences between genotype and phenotype that are associated to a potential solution. The genotype for a CGP chromosome have always same "structure": a vector with a length equal to the sum of the cells number times to the inputs number for a node and the outputs for the system. But the phenotype can have same length or a short length (because not all the nodes are evaluated for establishing the quality for a chromosome).

For each node it is need to specify the input(s) an the function(s) associated. We use integer numbers from $[0..p]$ range for coding the inputs (p = number of system inputs + the number of cells before current cell, if the columns graph are reading top down) and negative integer numbers from [-r..0] range for coding the functions (where r represents how many functions (components) can be used by the system).

CGP chromosomes are encoded as strings by reading the graph columns top down and printing the input nodes and the function symbol for each node. For instance, we can have a system with five different components (r = 5); each of these components can need one, two or three input parameters and provide a single output (figure 1). The system (chromosome) has four main inputs (i_0, i_1, i_2 and i_3) and twp outputs (o_0 and o_1). Therefore, each cell (node) of the graph can have three inputs and one output. We have coded with integer positive number the input indexes and with integer negative numbers the function indexes (-1 means the first function (component), -2 the second function, -3 the third function, -4 the fourth function and -5 the fifth function - first three functions have an arity equal to two, the forth function has the arity equal to 1 and the last function has the arity equal to 3). For this example each chromosome is represented like a Cartesian graph with two line and three columns.

A possible chromosome can take the next form:

$$\underbrace{0\ 1\ 2\ -1}_{i_4}\ \underbrace{2\ 3\ 1\ -2}_{i_5}\ \underbrace{4\ 5\ 3\ -3}_{i_6}\ \underbrace{4\ 3\ 5\ -4}_{i_7}\ \underbrace{6\ 1\ 5\ -4}_{i_8}\ \underbrace{4\ 5\ 7\ -5}_{i_9}\ \underbrace{6}_{o_0}\ \underbrace{9}_{o_1}$$

Decoding this chromosome we obtain:

$o_0 = i_6 = f_3(i_4, i_5) = f_3(f_1(i_0, i_1), f_2(i_2, i_3))$.

$o_1 = i_9 = f_5(i_4, i_5, i_7) = f_5(f_1(i_0, i_1), f_2(i_2, i_3), f_4(i_4)) = f_5(f_1(i_0, i_1), f_2(i_2, i_3), f_4(f_1(i_0, i_1)))$.

3.2 Algorithm

The algorithm used for evolving a component based system structure is described in this section. We use steady-state evolutionary model as underlying mechanism for our CGP implementation. The CGP algorithm starts by creating a random population of individuals (graphs). The following steps are repeated until a given number of generations is reached: Two parents are selected using a standard selection procedure. The parents are considered for mutation. The best offspring O replaces the worst individual W in the current population if O is better than W. Note that CGP method does not use crossover operator.

The mutation operator can act over any gene of CGP chromosome. During the mutation process, three situations can be found: *input cell mutation* (one or more inputs of a graph node is (are) changed with other initial inputs or with the output of a node placed on a previous column (relative to the column of current node)), *function mutation* (one o more cells can change their associated function (with another function from F)) and *output mutation* (one

or more system output can be changed; the new associated value(s) can be the output(s) of any node from graph (we don't restrict the inter-connectivity level to a threshold)).

3.3 Fitness measure

CGP uses a special kind of fitness assignment. The evaluation is performed by reading the chromosome only once and storing partial results by using dynamic programming [1].
Thus, the fitness of a CGP chromosome may be computed using the formula: $f = \sum_{j=1}^{N} |E_j - O_j|$, where N is the number of fitness cases, O_j is the value returned (for th jth fitness case) and E_j is the expected value for the fitness case j.

4 Experiments

4.1 Experiment type 1

The first type of experiments uses a genotype-phenotype mapping that does require all nodes to be connected to each other and the final solution doesn't contain duplicate components.
For this experiment we want to evolve a system with two inputs (two integer numbers), one output (an integer number) and four components: the first component is the digit sum for an integer number (*Dig_Sum* index -1), the second component is the digit product for an integer number (*Dig_Prod* index -2), the third component computes the sum of two integer numbers (*Sum* index -3) and the fourth component determines the product of two integer numbers (*Product* index -4).
The system (chromosome) computes the product between second number and the sum of digit sum of first number and digits product of second number:

$$o = i_1 * (Dig\_Sum(i_0) + Dig\_Prod(i_1)).$$

For this purpose we use a CGP algorithm with 100 individuals that are selected and mutated during 100 generations. At one step two chromosomes are chosen (using binary tournament selection procedure), than they are mutated and, finally, the best offspring replaces the worst individual from the population (if the best offspring is better than the worst individual from population). For this experiment each chromosome is represented like a Cartesian graph with one line and four columns and each cell (node) of the graph can have two inputs and one output.
A possible chromosome can take the next form:

$$\underbrace{1\ 0\ \text{-}1}_{i_2}\ \underbrace{0\ 1\ \text{-}2}_{i_3}\ \underbrace{2\ 3\ \text{-}2}_{i_4}\ \underbrace{4\ 1\ \text{-}4}_{i_5}\ \underbrace{5}_{o_0}.$$

Decoding this chromosome with two inputs i_0 and i_1 we obtain:

$$o = i_5 = Product(i_4, i_1) = Product(Dig\_Prod(i_2), i_1) = Product(Dig\_Prod(Dig\_Sum(i_1)), i_1).$$

The best chromosome encodes the optimal configuration:

$$\underbrace{1\ 0\ \text{-}2}_{i_2}\ \underbrace{0\ 1\ \text{-}1}_{i_3}\ \underbrace{2\ 3\ \text{-}3}_{i_4}\ \underbrace{4\ 1\ \text{-}4}_{i_5}\ \underbrace{5}_{o_0}$$

Decoding this chromosome we obtain:

$$o = i_5 = Product(i_4, i_1) = Product(Sum(i_2, i_3), i_1) = Product(Sum(Dig\_Prod(i_1), Dig\_Sum(i_0)), i_1).$$

4.2 Experiment type 2

The second type of experiments uses a genotype-phenotype mapping that does require all nodes to be connected to each other, and we have multiplicity of components in the grid.
Experiment 1. For this experiment we want to evolve a system with one input (an integer number), one output (an integer number) and three components: the first component is the inverse for an integer number, the second component is the second power for an integer number and the third component verifies if two numbers are equal. The system verifies if the inverse square of the given number is equal to the square inverse of the same number. For instance, if the number is 13 then $inverse(square(13)) = inverse(169) = 961$ is equal to $961 = square(31) = square(inverse(13))$.
Each chromosome is represented as a Cartesian graph with two lines and three columns. Each cell (node) of the graph can have two inputs and one output. For this purpose we use a CGP algorithm with 200 individuals that are selected and mutated during 30 generation. Mutation probability was 0.5.

The best chromosome encodes the optimal configuration:

$$\underbrace{0\,0\,\text{-}1}_{i_1}\ \underbrace{0\,0\,\text{-}2}_{i_2}\ \underbrace{2\,2\,\text{-}2}_{i_3}\ \underbrace{3\,2\,\text{-}1}_{i_4}\ \underbrace{1\,1\,\text{-}2}_{i_5}\ \underbrace{4\,5\,\text{-}3}_{i_6}\ \underbrace{7}_{o_0}$$

Decoding this chromosome we obtain:

$o = i_7 = Equal(i_4, i_5) = Equal(Square(i_2), Inverse(i_3))$
$= Equal(Square(Inverse(i_0)), Inverse(Square(i_0)))$.

Experiment 2. For this experiment we want to evolve a system with two inputs (two integer numbers), one output (an integer number) and four components: the first component is the sum of digits from odd positions for an integer number (*Dig_Sum_Odd* index -1), the second component is the product of digits from even positions for an integer number (*Dig_Prod_Even* index -2), the third component computes the sum of two integer numbers (*Sum* index -3)and the fourth component computes the difference between two integer numbers ($Diff$ index -4).

The system (chromosome) has two main inputs (two integer numbers) and computes the sum between first number and the substraction of the sum of digits from odd positions for the first number and the product of digits from even positions for the second number of the system.

Each chromosome is represented as a Cartesian graph with two lines and three columns. Each cell (node) of the graph can have two inputs and one output.

Running the same algorithm with those described in Section 4.1, we obtain two different chromosomes that can represents the optimal structure of our system:

$$\underbrace{1\,1\,\text{-}2}_{i_2}\ \underbrace{0\,1\,\text{-}1}_{i_3}\ \underbrace{0\,2\,\text{-}4}_{i_4}\ \underbrace{3\,2\,\text{-}3}_{i_5}\ \underbrace{3\,4\,\text{-}3}_{i_6}\ \underbrace{4\,2\,\text{-}4}_{i_7}\ \underbrace{6}_{o_0}$$

Decoding this chromosome we obtain:

$o = i_6 = Sum(i_3, i_4) = Sum(Dig\_Sum\_Odd(i_0), Diff(i_0, i_2))$
$= Sum(Dig\_Sum\_Odd(i_0), Diff(i_0, Dig\_Prod\_Even(i_1)))$
$= Dig\_Sum\_Odd(i_0) + (i_0 - Dig\_Prod\_Even(i_1))$.

Note that the results of cells that corresponds to i_5, respectively to i_7 are not used.

$$\underbrace{1\,0\,\text{-}2}_{i_2}\ \underbrace{0\,1\,\text{-}1}_{i_3}\ \underbrace{3\,2\,\text{-}4}_{i_4}\ \underbrace{0\,2\,\text{-}3}_{i_5}\ \underbrace{4\,2\,\text{-}4}_{i_6}\ \underbrace{0\,4\,\text{-}3}_{i_7}\ \underbrace{7}_{o_0}$$

Decoding this chromosome we obtain:

$o = i_7 = Sum(i_0, i_4) = Sum(i_0, Diff(i_3, i_2))$
$= Sum(i_0, Diff(Dig\_Sum\_Odd(i_0), Dig\_Prod\_Even(i_1)))$
$= i_0 + (Dig\_Sum\_Odd(i_0) - Dig\_Prod\_Even(i_1))$.

Note that the results of cells that corresponds to i_5, respectively to i_6 are not used.

Experiment 3. For this experiment we use the example from Figure 1.b). Each chromosome is represented as a Cartesian graph with three lines and three columns. Each cell (node) of the graph can have three inputs and one output. For this purpose we use a CGP algorithm with 200 individuals that are selected and mutated during 130 generations.

The best chromosome encodes the optimal configuration:

$$\underbrace{2\,3\,0\,\text{-}2}_{i_4}\ \underbrace{1\,2\,3\,\text{-}3}_{i_5}\ \underbrace{0\,1\,3\,\text{-}1}_{i_6}\ \underbrace{5\,1\,0\,\text{-}2}_{i_7}\ \underbrace{4\,6\,4\,\text{-}3}_{i_8}\ \underbrace{4\,0\,5\,\text{-}4}_{i_9}\ \underbrace{6\,9\,4\,\text{-}5}_{i_{10}}\ \underbrace{8\,0\,9\,\text{-}3}_{i_{11}}\ \underbrace{1\,7\,4\,\text{-}2}_{i_{12}}\ \underbrace{8}_{o_0}\ \underbrace{10}_{o_1}$$

5 Summary and Conclusions

In this paper we have introduce a new method to develop component-based systems using Cartesian Genetic Programming. Given a set of components that are integrated into the system we evolved the solution, taking into account the component composition behaviour. We obtain also different results for the same system, due to the associativity and distributivity of addition and substraction.

There are some characteristics of CGP that permit to use it for evolving component-based system structures: a chromosome can have one or more outputs, the problem's outputs are subject to evolution and each function associated to a node can use all the node inputs or only that inputs that it needs.

References

[1] R. Bellman, *Dynamic Programming*, Princeton University Press, New Jersey, 1957.

[2] I. Crnkovic, M. Larsson, *Building reliable component-based software systems*, Artech House, 2002

[3] A. M. Gravell, and P. Henderson, Executing formal specifications need not be harmful, *Software Engineering Journal*, 11(2):104-110, IEE/BCS, March 1996

[4] C. A. R. Hoare, The role of formal techniques: past, current and future or how did software get so reliable without proof?, *18th International Conference on Software Engineering (ICSE-18)*, Berlin, IEEE Computer Society Press, 1996, pp. 233-234.

[5] J. F. Miller, P. Thomson, and T. Fogarty, *Designing Electronic Circuits using Evolutionary Algorithms. Arithmetic Circuits: A Case Study*. In Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, D. Quagliarella, J. Periaux, C. Poloni and G. Winter (Editors), pp. 105-131, Chechester, UK-Wiley, 1997.

[6] J. F. Miller and P. Thomson, Aspects of Digital Evolution: Evolvability and Architecture. In *Proceedings of the Parallel Problem Solving from Nature*, V, A. E. Eiben, T. Bäck, M. Schoenauer, and H-P Schwefel (Editors), pp. 927-936, Springer, 1998.

[7] J. F. Miller. An Empirical Study of the Efficiency of Learning Boolean Functions using a Cartesian Genetic Programming Approach. *In Proceedings of the 1st Genetic and Evolutionary Computation Conference*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, (Editors), Vol. 2, pp. 1135-1142, Morgan Kaufmann, San Francisco, CA, 1999.

[8] Julian F. Miller and Peter Thomson, Cartesian Genetic Programming, *Proceedings of the European Conference on Genetic Programming*, Springer-Verlag, 2000, pp. 121-132.

[9] Szyperski C., *Component Software, Beyond Object-Oriented Programming,* ACM Press, Addison-Wesley, NJ,1998.

[10] R. J. Walters, *A Graphically based language for constructing, executing and analyzing models of software systems*, PhD, December 2002.

Andreea Vescan, Laura Dioşan
Babeş-Bolyai University
Faculty of Mathematics and Computer Science
Computer Science Department
M. Kogalniceanu nr. 1, RO-400084 Cluj-Napoca, Romania
E-mail: {afanea, lauras}@cs.ubbcluj.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 486-491

Modular Analysis of Concurrency in Petri Nets

Cristian Vidraşcu

Abstract: The goal of this paper is to study the relationships between the concurrency-degrees of a Petri net and those of its subnets.

Keywords: models of parallel/distributed systems, Petri nets, concurrency, modular analysis.

1 Introduction

A Petri net is a mathematical model used for the specification and the analysis of parallel/distributed systems. An introduction about Petri nets can be found in [1]. The concurrency-degrees are a measurement of the concurrency in Petri nets, which was first introduced in [2]. It is useful to introduce a measure of concurrency for parallel/distributed systems, for answering a question like this: What is the meaning of the fact that in the system S_1 the concurrency is greater than in the system S_2?
The problem of concurrency has been studied for Petri nets, but, since they are used as suitable models for real-world parallel or distributed systems, the results have been applicable also to these systems. The basic idea is that the number of transitions which can fire simultaneously in a Petri net which models a real system, can be used as an intuitive measure of the concurrency of that system.
An advantage of studying the concurrency-degrees is that they can be computed for the model during the design of a system, and this will usually lead to an improved design.
This paper studies the relationships between the concurrency-degrees of a Petri net and those of its subnets. The remainder of this paper is organized as follows. Section 2 presents the basic terminology and notation concerning Petri nets. In Section 3 we present the results concerning the modular analysis of concurrency-degrees for Petri nets. Section 4 concludes this paper and formulates some open problems.

2 Preliminaries

We will assume to be known the basic terminology and notation about sets, relations and functions, vectors, multi-sets and formal languages. Let us just briefly remind that a *multi-set m*, over a non-empty set *S*, is a function $m : S \to \mathbb{N}$, usually represented as a formal sum: $\sum_{s\in S} m(s)`s$. It will be sometimes identified with a $|S|$-dimensional vector. The operations and relations on multi-sets are defined component-wise. $S_{MS}$ denotes the set of all multi-sets over *S*. The empty multi-set $\sum_{s\in S} 0`s$ is denoted by $\emptyset$. The *size* of the multi-set *m* is defined as $|m| = \sum_{s\in S} m(s)$, and *m* is called *infinite* iff $|m| = \infty$.
This section will establish the basic terminology and results concerning Petri nets in order to give the reader the necessary prerequisites to understand this paper (for details the reader is referred to [1]).

2.1 Petri Nets

Definition 1. A *Place/Transition net*, shortly *P/T-net*, (finite, with infinite capacities), abbreviated *PTN*, is a 4-tuple $\Sigma = (S,T,F,W)$, where *S* and *T* are two finite non-empty sets (of *places* and *transitions*, resp.), $S \cap T = \emptyset$, $F \subseteq (S \times T) \cup (T \times S)$ is the *flow relation* and $W : (S \times T) \cup (T \times S) \to \mathbb{N}$ is the *weight function* of Σ satisfying $W(x,y) = 0$ iff $(x,y) \notin F$.

Definition 2. A *marking* of a *PTN* Σ is a function $M : S \to \mathbb{N}$, i.e. a multiset over S ; it will be sometimes identified with a $|S|$-dimensional vector. The operations and relations on vectors are componentwise defined. $\mathbb{N}^S$ denotes the set of all markings of Σ.

Definition 3. A *marked PTN*, abbreviated *mPTN*, is a pair $\gamma = (\Sigma, M_0)$, where Σ is a *PTN* and M_0, called the *initial marking* of γ, is a marking of Σ.

In the sequel we often use the term “Petri net"(*PN*) or “net" whenever we refer to a *PTN* (*mPTN*) and it is not necessary to specify its type (i.e. marked or unmarked).

Notation 4. Let Σ be a Petri net, $t \in T$ and $w \in T^*$. The functions $t^-, t^+ : S \to \mathbb{N}$ and $\Delta t, \Delta w : S \to \mathbb{Z}$ are defined by $t^-(s) = W(s,t)$, $t^+(s) = W(t,s)$, $\Delta t(s) = t^+(s) - t^-(s)$, and
$\Delta w(s) = \begin{cases} 0 & \text{, if } w = \lambda \\ \sum_{i=1}^{n} \Delta t_i(s) & \text{, if } w = t_1 t_2 \ldots t_n \ (n \geq 1) \end{cases}$, for all $s \in S$.

Definition 5. The sequential behaviour of a net Σ is given by the *firing rule*, which consists of :
i) the *enabling rule*: a transition t is *enabled* at a marking M in Σ (or t is *fireable* from M), abbreviated $M[t\rangle_\Sigma$, iff $t^- \leq M$; ii) the *computing rule*: if $M[t\rangle_\Sigma$, then t may *occur* yielding a new marking M', abbreviated $M[t\rangle_\Sigma M'$, defined by $M' = M + \Delta t$.

In fact, for any transition t of Σ we have a binary relation on $\mathbb{N}^S$, denoted by $[t\rangle_\Sigma$ and given by: $M[t\rangle_\Sigma M'$ iff $t^- \leq M$ and $M' = M + \Delta t$. If $t_1, t_2, \ldots, t_n$, $n \geq 1$, are transitions of Σ, $[t_1 t_2 \ldots t_n\rangle_\Sigma$ will denote the classical product of the relations $[t_1\rangle_\Sigma, \ldots, [t_n\rangle_\Sigma$. Moreover, we also consider the relation $[\lambda\rangle_\Sigma$ given by $[\lambda\rangle_\Sigma = \{(M,M) | M \in \mathbb{N}^S\}$.

Definition 6. Let $\gamma = (\Sigma, M_0)$ be a *mPTN*, and $M \in \mathbb{N}^S$. The word $w \in T^*$ is called a *transition sequence* from M in Σ if there exists a marking M' of Σ such that $M[w\rangle_\Sigma M'$. Moreover, the marking M' is called *reachable* from M in Σ.

Notation 7. $TS(\Sigma, M) = \{w \in T^* \,|\, M[w\rangle_\Sigma\}$ denotes the set of all transition sequences from M in Σ, and $RS(\Sigma, M) = \{M' \in \mathbb{N}^S | \exists w \in TS(\Sigma, M) : M[w\rangle_\Sigma M'\}$ the set of all reachable markings from M in Σ. In the case $M = M_0$, the set $TS(\Sigma, M_0)$ is abbreviated by $TS(\gamma)$, and the set $RS(\Sigma, M_0)$ is abbreviated by $RS(\gamma)$ or by $[M_0\rangle_\gamma$ and it is called the *reachability set* of γ.

Concurrency-degrees for Petri Nets

The concurrency-degrees are a measurement of the concurrency in Petri nets, which was first introduced in [2] for classical P/T-nets. A more appropriate definition of concurrency-degrees for them, which takes into consideration the auto-concurrency (i.e the case of the transitions concurrently enabled with themselves), was given in [3], by considering the notion of a step fireable at a marking as a multiset of transitions. Also, a finer notion was introduced in [3], namely the concurrency-degrees w.r.t. a set of transitions, and the computability of concurrency-degrees was studied.
In the sequel, we will present these definitions.
Let $\Sigma = (S, T, F, W)$ be a P/T-net, M a marking of Σ, and $T' \subseteq T$ a subset of transitions of Σ.

Definition 8. A *step* Y is a non-empty and finite multiset over T. A *T'-step* Y is a step satisfying $Y(t) = 0$, for all $t \in T - T'$ (practically, Y is a non-empty and finite multiset over the subset T').

Definition 9. A step Y is *enabled* at the marking M in Σ (or Y is *fireable* from M), and we say also that Y is a multiset of transitions *concurrently enabled* at M, abbreviated $M[Y\rangle_\Sigma$, iff $\sum_{t \in T} Y(t) \cdot t^- \leq M$. Moreover, if Y is enabled at M in Σ, then by occuring the step Y at the marking M it is produced the marking $M' = M + \sum_{t \in T} Y(t) \cdot \Delta t$, abbreviated $M[Y\rangle_\Sigma M'$.

Definition 10. The *concurrency-degree w.r.t. T' of Σ at M* is defined as the maximum (i.e. the supremum) number of (not necessarily distinct) transitions from T' which are concurrently enabled at M:

$$d(\Sigma, T', M) = \sup\{\, |Y| \mid Y \text{ is a } T'\text{-step enabled at } M \text{ in } \Sigma\}\,.$$

In the case $T' = T$, the degree $d(\Sigma, T', M)$ is abbreviated by $d(\Sigma, M)$ and it is called the *concurrency-degree of Σ at the marking M*.

Remark 1. i) The concurrency-degree (w.r.t. T') at a marking M of a Petri net Σ represents the supremum number of transitions (from T') concurrently enabled at the marking M in Σ.
ii) $d(\Sigma, T', M) = +\infty$ iff there exists a transition $t \in T'$ with $t^- = 0$.

Let $\gamma = (\Sigma, M_0)$ be a marked P/T-net, and $T' \subseteq T$ a subset of transitions of Σ.

Definition 11. The *inferior and superior concurrency-degree w.r.t. T' of γ* are defined as the minimum, and resp. the maximum (i.e. the supremum), number, at any moment, of (not necessarily distinct) transitions from T' which are maximal concurrently enabled:

$$d^-(\gamma,T') = \min\{\, d(\Sigma,T',M) \,|\, M \in [M_0\rangle_\gamma\} \quad \text{and} \quad d^+(\gamma,T') = \sup\{\, d(\Sigma,T',M) \,|\, M \in [M_0\rangle_\gamma\}\,.$$

If $d^-(\gamma,T') = d^+(\gamma,T')$, then this number is called the *concurrency-degree w.r.t. T' of γ* and it is denoted by $d(\gamma,T')$. Moreover, in the case $T' = T$, the degrees $d^-(\gamma,T')$, $d^+(\gamma,T')$, and $d(\gamma,T')$ are abbreviated by $d^-(\gamma)$, $d^+(\gamma)$, and $d(\gamma)$ respectively, and "*w.r.t. T'*" is dropped from their names.

Remark 2. i) $d^-(\gamma,T')$ means that at any reachable marking M of γ there exist at least $d^-(\gamma,T')$ transitions (from T') concurrently enabled at M.
ii) $d^+(\gamma,T')$ means that at any reachable marking M of γ there exist at most $d^+(\gamma,T')$ transitions (from T') concurrently enabled at M.
iii) $d(\gamma,T')$ means that at any reachable marking M of γ there exist $d(\gamma,T')$ transitions (from T') concurrently enabled at M, and there is no reachable marking M' of γ with more than $d(\gamma,T')$ transitions (from T') concurrently enabled at M'.

For more comments about the intuitive meaning of concurrency-degrees for P/T-nets, and the results regarding their computability, the reader is referred to [3].

3 Modular Analysis of Concurrency

We can take into consideration the problem of modularization for Petri nets: a P/T-net can be "decomposed" into several modules, i.e. subnets of it, which have in common some locations of the net; these locations play the role of "interface" (i.e., they are *shared*) between two or more modules. Using this setting, the study of the concurrency in the global net can be done by analysing the concurrency of the subnets which form that net.
Therefore, it is useful to study the relationships between the concurrency-degrees of a global net and those of the subnets which compose that net.
The following result shows the relationship which exists between the concurrency-degree w.r.t. the union of two disjoint sets of transitions and the concurrency-degrees w.r.t. each of those two sets:

Theorem 12. i) Let Σ be a P/T-net, $T_1, T_2 \subseteq T$ two disjoint sets of transitions of Σ, and M an arbitrary marking of Σ. Then the following inequality holds:

$$d(\Sigma,T_1 \cup T_2,M) \leq d(\Sigma,T_1,M) + d(\Sigma,T_2,M)\ . \tag{1}$$

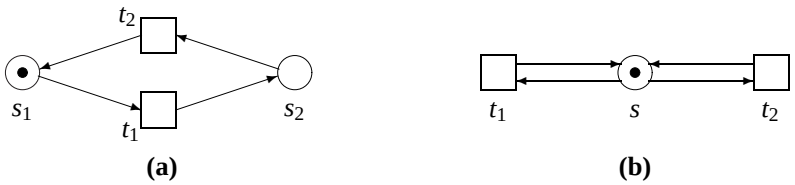
ii) Let γ be a marked P/T-net, and $T_1, T_2 \subseteq T$ two disjoint sets of transitions of γ. Then we have

$$d^+(\gamma,T_1 \cup T_2) \leq d^+(\gamma,T_1) + d^+(\gamma,T_2)\ . \tag{2}$$

Proof. i) Let Σ be a *PTN*, $T_1, T_2 \subseteq T$ two disjoint sets of transitions of Σ, and M an arbitrary marking of Σ. Moreover, let Y be an arbitrary $(T_1 \cup T_2)$-step enabled at M in Σ. Thus, $M[Y\rangle_\Sigma$, and since $T_1 \cap T_2 = \emptyset$, by the step enabling rule of Definition 9 it follows that

$$M \geq \sum_{t \in T_1 \cup T_2} Y(t) \cdot t^- = \sum_{t \in T_1} Y(t) \cdot t^- + \sum_{t \in T_2} Y(t) \cdot t^-\,,$$

so we can conclude that $\sum_{t \in T_1} Y(t) \cdot t^- \leq M$ and, respectively, $\sum_{t \in T_2} Y(t) \cdot t^- \leq M$.
First inequality means that the step denoted by $Y|_{T_1} \stackrel{not}{=} \sum_{t \in T_1} Y(t)`t$ is fireable at $M$ in $\Sigma$, i.e. $M[Y|_{T_1}\rangle_\Sigma$. By Definition 10, this means that $|Y|_{T_1}| \leq d(\Sigma,T_1,M)$. Similarly, from the second inequality from above we deduce that the step denoted by $Y|_{T_2} \stackrel{not}{=} \sum_{t \in T_2} Y(t)`t$ is a T_2-step enabled at M in Σ, and therefore we have that $|Y|_{T_2}| \leq d(\Sigma,T_2,M)$.
Since $Y = Y|_{T_1} + Y|_{T_2}$, we can conclude that $|Y| = |Y|_{T_1}| + |Y|_{T_2}| \leq d(\gamma,T_1,M) + d(\gamma,T_2,M)$.
Thus, we showed that $|Y| \leq d(\gamma,T_1,M) + d(\gamma,T_2,M)$, for all Y which are $(T_1 \cup T_2)$-steps enabled at M in Σ. By taking the supremum in this inequality, after Y as a $(T_1 \cup T_2)$-step enabled at M in Σ, and using Definition 10, we obtain the desired inequality (1).



(a) **(b)**

Figure 1: The P/T-nets from Example 1

ii) Let $\gamma=(\Sigma,M_0)$ be a *mPTN*, and $T_1,T_2\subseteq T$ two disjoint sets of transitions of γ. By Definition 11, we have that $d(\gamma,T_1,M)\leq d^+(\gamma,T_1)$ and $d(\gamma,T_2,M)\leq d^+(\gamma,T_2)$, for any reachable marking M of γ.
Since, by pct. i), the inequality (1) holds for any arbitrary marking M of Σ, and, therefore, it holds particularly for the reachable ones, we conclude that $d(\Sigma,T_1\cup T_2,M)\leq d^+(\Sigma,T_1)+d^+(\Sigma,T_2)$, for all $M\in[M_0\rangle_\gamma$. By taking the supremum after $M\in[M_0\rangle_\gamma$ in this inequality, and using Definition 11, we obtain the desired inequality (2). □

Remark 3. Unfortunately, regarding the inferior concurrency-degree, neither an inequality like (2), nor one with an inverse sign, holds true. This remark is justified by the following counterexample.
Example 1. Let $\gamma_1=(\Sigma_1,M_0)$ be the marked net from Fig. 1 (a). It is easy to see that its reachability set is $[M_0\rangle_{\gamma_1}=\{M_0,M_1\}$, where $M_0=(1,0)$ and $M_1=(0,1)$.Also, $\{t_1\}$ is the only step enabled at M_0 in Σ_1, and $\{t_2\}$ is the only step enabled at M_1 in Σ_1. Thus, the inferior concurrency-degrees w.r.t. the various sets of transitions are as follows: $d^-(\gamma_1,\{t_1\})=0$, $d^-(\gamma_1,\{t_2\})=0$, and $d^-(\gamma_1,\{t_1,t_2\})=1$.
Therefore, in this case we have the strict inequality $d^-(\gamma,\{t_1,t_2\})>d^-(\gamma,\{t_1\})+d^-(\gamma,\{t_2\})$.
Now, let us consider the marked net $\gamma_2=(\Sigma_2,M_0)$ from Fig. 1 (b). We have $[M_0\rangle_{\gamma_2}=\{M_0\}$, where $M_0=(1)$, and $\{t_1\}$, $\{t_2\}$ are the only steps enabled at M_0 in Σ_2. The inferior concurrency-degrees w.r.t. the various sets of transitions are as follows: $d^-(\gamma_2,\{t_1\})=1$, $d^-(\gamma_2,\{t_2\})=1$, and $d^-(\gamma_2,\{t_1,t_2\})=1$.
Therefore, in this case we have the strict inequality $d^-(\gamma,\{t_1,t_2\})<d^-(\gamma,\{t_1\})+d^-(\gamma,\{t_2\})$.

Despite Remark 3, which tell us that for the inferior concurrency-degree there exists no upper bound like the ones which exist for the concurrency-degree at a marking and for the superior concurrency-degree, we can still specify a lower bound for the inferior concurrency-degree of Petri nets, namely:

Theorem 13. Let γ be a marked Petri net, and $T_1,T_2\subseteq T$ two disjoint sets of transitions of γ. Then

$$d^-(\gamma,T_1\cup T_2)\geq\max\{d^-(\gamma,T_1),d^-(\gamma,T_2)\} \ . \tag{3}$$

Proof. Let $\gamma=(\Sigma,M_0)$ be a *mPTN*, and $T_1,T_2\subseteq T$ two disjoint sets of transitions of γ.
Let M be an arbitrary marking of the net Σ. Obviously, any T_1-step enabled at the marking M in Σ is also a $(T_1\cup T_2)$-step enabled at M in Σ (being a multiset over $T_1\cup T_2$ having zero multiplicities for the elements from T_2). Therefore, by Definition 10, we can deduce that $d(\Sigma,T_1\cup T_2,M)\geq d(\Sigma,T_1,M)$, for any arbitrary marking M, and thus, particularly, also for any $M\in[M_0\rangle_\gamma$.
But, from Definition 11, it follows that $d(\Sigma,T_1,M)\geq d^-(\gamma,T_1)$, for all $M\in[M_0\rangle_\gamma$. Therefore, we can conclude that $d(\Sigma,T_1\cup T_2,M)\geq d^-(\gamma,T_1),\forall M\in[M_0\rangle_\gamma$.
Similarly it can be shown that $d(\Sigma,T_1\cup T_2,M)\geq d^-(\gamma,T_2),\forall M\in[M_0\rangle_\gamma$.
Using these last two inequalities, we deduce that $d(\Sigma,T_1\cup T_2,M)\geq\max\{d^-(\gamma,T_1),d^-(\gamma,T_2)\}$, for all $M\in[M_0\rangle_\gamma$.
By taking the minimum after $M\in[M_0\rangle_\gamma$ in this inequality, and using Definition 11, we obtain the desired inequality (3). □

Remark 4. The inequalities (1) and (2) from Theorem 12, as well as the inequality (3) from Theorem 13, hold also for the generalized case of any finite union of pairwise-disjoint sets of transitions. (This can be easily proved by applying these inequalities, reiteratively for unions of two disjoint sets of transitions.)

A particular case of these generalized inequalities for Petri nets is represented by the case when the sets of transitions are all singletons (i.e., each set has only one element). In this case we obtain the following result, which expresses the relationship between the concurrency-degree w.r.t. a set of transitions and the concurrency-degrees w.r.t. each individual transition from that set:

Corollary 14. i) Let Σ be a Petri net, $T'\subseteq T$ a set of transitions of Σ, and M any marking of Σ. Then

$$d(\Sigma,T',M)\leq\sum_{t\in T'}d(\Sigma,\{t\},M) \ . \tag{4}$$

ii) Let γ be a marked Petri net, and $T'\subseteq T$ a set of transitions of γ. The following inequalities hold:

$$d^+(\gamma,T') \leq \sum_{t\in T'} d^+(\gamma,\{t\}) \quad \text{and} \quad d^-(\gamma,T') \geq \max_{t\in T'} d^-(\gamma,\{t\}) \ . \tag{5}$$

Proof. These inequalities follow as simple consequences from Theorem 12 and 13, by applying reiteratively (by $|T'|-1$ times) the corresponding inequalities from the two mentioned theorems. □

Based on Example 1, an interesting question may arise, namely: when it holds with equality any of the inequalities (1), (2) and (3), or the analogous of inequality (2) for the inferior concurrency-degree ?
Let us notice the following fact: if Σ is a Petri net, $T' \subseteq T$ a set of transitions of Σ, and M_1, M_2 are two arbitrary markings of Σ such that $M_1(s) = M_2(s)$, for any location $s \in {}^\bullet T'$, then any T'-step enabled at M_1 in Σ is also enabled at M_2 in Σ and vice versa. Thus, the set of T'-steps enabled at M_1 in Σ is equal with the set of T'-steps enabled at M_2 in Σ, and, therefore, $d(\Sigma,T',M_1) = d(\Sigma,T',M_2)$.
In other words, the concurrency-degree at a marking w.r.t. a set of transitions depends *only* on the components of that marking which correspond to the pre-locations of the transitions from that set.
This remark gives us a structural property of a Petri net which is a sufficient condition for some of the above mentioned equalities to hold:

Theorem 15. Let $\Sigma = (S,T,F,W)$ be a P/T-net, and $T_1, T_2 \subseteq T$ two disjoint sets of transitions of Σ. If Σ satisfies the property ${}^\bullet T_1 \cap {}^\bullet T_2 = \emptyset$ (in other words, for any pair of transitions $t_1 \in T_1$ and $t_2 \in T_2$, t_1 and t_2 do not have common pre-locations), then for any arbitrary marking M of Σ we have the equality:

$$d(\Sigma,T_1 \cup T_2,M) = d(\Sigma,T_1,M) + d(\Sigma,T_2,M) \ . \tag{6}$$

Moreover, if we consider an initial marking for the net Σ, i.e. let $\gamma = (\Sigma, M_0)$ with Σ as above, then regarding the inferior concurrency-degree the following inequality holds:

$$d^-(\gamma,T_1 \cup T_2) \geq d^-(\gamma,T_1) + d^-(\gamma,T_2) \ . \tag{7}$$

Proof. By inequality (1) from Theorem 12, for proving (6) it is sufficient to show that we have the inequality (1) with an inverse sign in the hypothesis ${}^\bullet T_1 \cap {}^\bullet T_2 = \emptyset$ satisfied by the net Σ.
Let Y_1 be an arbitrary T_1-step enabled at the marking M in Σ, and Y_2 an arbitrary T_2-step enabled at M in Σ. Then, by the step enabling rule of Definition 9, it follows that

$$Y_1^-(s) \stackrel{not}{=} \sum_{t\in T_1} Y(t)\cdot t^-(s) \leq M(s) \quad \text{and} \quad Y_2^-(s) \stackrel{not}{=} \sum_{t\in T_2} Y(t)\cdot t^-(s) \leq M(s), \text{ for all } s \in S.$$

Let $Y = Y_1 + Y_2$. Thus, Y is a $(T_1 \cup T_2)$-step. Then $Y^-(s) \stackrel{not}{=} \sum_{t\in T_1\cup T_2} Y(t)\cdot t^-(s) = Y_1^-(s) + Y_2^-(s)$, for all $s \in S$.
Using the fact that $t^-(s) = 0$ iff $s \notin {}^\bullet t$, for any $t \in T$ and $s \in S$, and since, by hypothesis, ${}^\bullet T_1 \cap {}^\bullet T_2 = \emptyset$, it follows that for any $s \in S$, one and only one of the following three cases is possible:
(i) $s \in {}^\bullet T_1$. Then $s \notin {}^\bullet T_2$ and, therefore, $Y_2^-(s) = 0$. So, we obtain that $Y^-(s) = Y_1^-(s) \leq M(s)$.
(ii) $s \in {}^\bullet T_2$. Then $s \notin {}^\bullet T_1$ and, therefore, $Y_1^-(s) = 0$. So, we obtain that $Y^-(s) = Y_2^-(s) \leq M(s)$.
(iii) $s \in S - ({}^\bullet T_1 \cup {}^\bullet T_2)$. Then $s \notin {}^\bullet T_1$ and $s \notin {}^\bullet T_2$, and, therefore, $Y_1^-(s) = Y_2^-(s) = 0$. So, we obtain in this case that $Y^-(s) = 0 \leq M(s)$.
In conclusion, we proved that $Y^-(s) \leq M(s)$, for any $s \in S$. This inequality means that Y is a $(T_1 \cup T_2)$-step enabled at M in Σ. By using Definition 10, it follows that $|Y| \leq d(\Sigma,T_1 \cup T_2,M)$. Thus, since $Y = Y_1 + Y_2$, we have that $|Y_1| + |Y_2| = |Y| \leq d(\Sigma,T_1 \cup T_2,M)$.
Therefore, we proved that $d(\Sigma,T_1 \cup T_2,M) \geq |Y_1| + |Y_2|$, for any arbitrary T_1-step enabled at M in Σ, Y_1, and any arbitrary T_2-step enabled at M in Σ, Y_2.
By succesively taking the supremum in the above inequality, after Y_1 as a T_1-step enabled at M in Σ, and then after Y_2 as a T_2-step enabled at M in Σ, and using Definition 10, we obtain the desired inequality:

$$d(\Sigma,T_1 \cup T_2,M) \geq d(\Sigma,T_1,M) + d(\Sigma,T_2,M) \ .$$

To prove the second part of this theorem, let $M \in [M_0\rangle_\gamma$ be an arbitrary reachable marking. Then, by Definition 11, we have that $d(\Sigma,T_1,M) \geq d^-(\gamma,T_1)$ and $d(\Sigma,T_2,M) \geq d^-(\gamma,T_2)$. Thus, by applying the first part of this theorem, we obtain that

$$d(\Sigma,T_1 \cup T_2,M) = d(\Sigma,T_1,M) + d(\Sigma,T_2,M) \geq d^-(\gamma,T_1) + d^-(\gamma,T_2).$$

Therefore, we proved that $d(\Sigma,T_1 \cup T_2,M) \geq d^-(\gamma,T_1) + d^-(\gamma,T_2)$, for all $M \in [M_0\rangle_\gamma$. By taking the minimum after $M \in [M_0\rangle_\gamma$ in this inequality, and using Definition 11, we obtain the inequality (7). □

Remark 5. Obviously, equality (6) and inequality (7) from Theorem 15 hold also for the generalized case of any finite union of pairwise-disjoint sets of transitions. (This remark can be easily proved by applying (6), and respectively (7), reiteratively for unions of two disjoint sets of transitions.)

A particular case of these generalized equality and inequality for Petri nets is represented by the case when the sets of transitions are all singletons. In this case we obtain the following result, which expresses a sufficient condition for having the above mentioned relationships between the concurrency-degree w.r.t. a set of transitions and the concurrency-degrees w.r.t. each individual transition from that set:

Corollary 16. Let $\Sigma = (S,T,F,W)$ be a P/T-net, and $T' \subseteq T$ a set of transitions of Σ. If Σ satisfies the property ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$, for any $t_1, t_2 \in T'$, then the following equality holds for any marking M of Σ:

$$d(\Sigma, T', M) = \sum_{t \in T'} d(\Sigma, \{t\}, M) \ . \tag{8}$$

Moreover, if we consider an initial marking for the net Σ, i.e. let $\gamma = (\Sigma, M_0)$ with Σ as above, then regarding the inferior concurrency-degree the following inequality holds:

$$d^-(\gamma, T') \geq \sum_{t \in T'} d^-(\gamma, \{t\}) \ . \tag{9}$$

Proof. These relations follow as simple consequences from Theorem 15, by applying reiteratively (by $|T'|-1$ times) the corresponding relations from that theorem. □

Remark 6. Unfortunately, the condition ${}^\bullet T_1 \cap {}^\bullet T_2 = \emptyset$ is not sufficient neither for having (2) with equality for the superior concurrency-degree, nor for having the similar equality for the inferior concurrency-degree.
This afirmation is justified by the net γ_1 from Fig. 1 (a), which satisfies the condition ${}^\bullet t_1 \cap {}^\bullet t_2 = \emptyset$, and for which we saw in Example 1 that $d^-(\gamma_1, \{t_1, t_2\}) = 1 > 0 + 0 = d^-(\gamma_1, \{t_1\}) + d^-(\gamma_1, \{t_2\})$. Also, for this net, we have that $d^+(\gamma, \{t_1, t_2\}) = 1 < 1 + 1 = d^+(\gamma, \{t_1\}) + d^+(\gamma, \{t_2\})$.
However, inequality (7), which holds in the hypothesis ${}^\bullet T_1 \cap {}^\bullet T_2 = \emptyset$, represents an improvement of the lower bound given by inequality (3), for the inferior concurrency-degree.

4 Summary and Conclusions

In this paper we have studied the relationships between the concurrency-degrees of a Petri net and those of its subnets.
Some problems remain to be studied, for example to find the conditions for which the inequality (2) holds true with equality for the superior and resp. the inferior concurrency-degree of a marked Petri net.

References

[1] W. Reisig, *Petri Nets. An Introduction, EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, Berlin, 1985.

[2] F.L. Ţiplea, T. Jucan, Şt. Dumbravă, "Modeling systems by Petri nets with different degrees of concurrency," *Proc. of the 4th International Symposium on Automatic Control and Computer Science – SACCS'93*, Technical Univesity "Gh. Asachi" of Iaşi, Romania, pp. 48–54, 1993.

[3] C. Vidraşcu, T. Jucan, "Concurrency-degrees for P/T-nets," *Scientific Annals of University "Al. I. Cuza" of Iaşi, Computer Science Section*, Tome XIII, pp. 91–103, 2003.

Cristian Vidraşcu
University "Al. I. Cuza" of Iaşi
Faculty of Computer Science
Address: 16, Gen. Berthelot St., 700483 - Iaşi, România
E-mail: vidrascu@infoiasi.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 492-496

Tourism Implications in Economic Growth. A Cybernetic Approach

Marian Zaharia, Rodica Manuela Gogonea

Abstract: Economists, mathematicians, statistics specialists have given rise to an actual theory of economic growth that makes up the background for the interdisciplinary theory of economic modeling, by founding and drawing up static or dynamic models, economic-mathematical or economic-environmental models. Among all these approaches, the tertiary or service field has become dominant in the post-war period, with the service market ranking ever more importantly, encompassing extremely varied activities, heterogeneous in terms of content, yet taking distinct manifest shapes. In the same time, in a systemic approach, the services field is a complex and dynamic cybernetic system with its own structure interconnected within the cybernetic system of the national/world economy.

Keywords: economic growth, economic modeling, cybernetic model, complex and dynamic systems

1 Introduction

Literature in the field currently encompasses the results of multiple researches that have laid the foundations of new interpretations of the concept nowadays known as Şthe new economic growthŤ or Şsustainable economic growthŤ. The theories and the various view points concerning the tourismŠs implications in the process of economic growth have been given starting with the implications and the considerations of the touristic activity under the influence of the touristic phenomenonŠs constraints and optimizations upon the economy. The economic importance of this field is reasoned by presenting tourism as a means of diversity of the economic structures, of turning to account the resource, of creator and user of national income and of new jobs, of investmentsŠ incentive, of factor of reducing the inflationist phenomenon, as well as a component of the external relations. The diversity of services, the seasonal character of tourism, the typology of the tourismŠs forms, the pricesŠ and the tariffsŠ category implies using an indicatorsŠ system, which can be approached as a cybernetic system. At its turn, this system is compounded by a multitude of subsystems with considerable implications to economic growth.

2 Economic modeling and growth premises

The modeling of tourism field is a part of the very complex problem the economic growth modeling process. The premises underlying the economic growth modeling process are laid down below, figure 1.

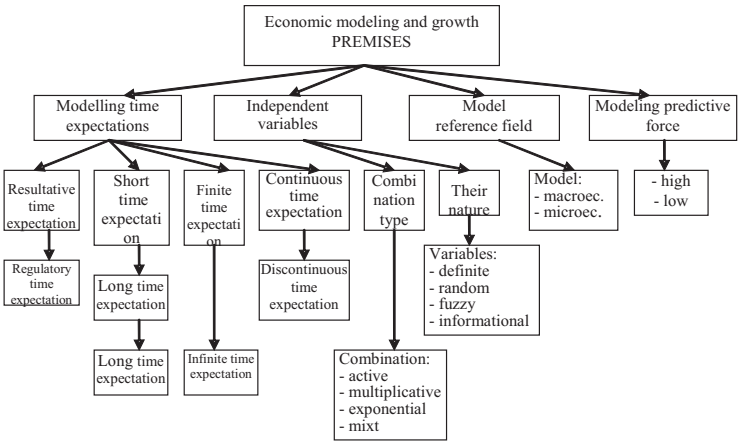


Figure 1: Economic modeling and growth premises

Choosing an economic growth model involves a careful scrutiny of all economic structure upgrade opportunities. This scrutiny is followed by the choice of variants that meet the conditions and restrictions resulting from the

chosen model, which has been deemed adequate to existing conditions. Setting up models, during the past years, has focused on a balanced economic growth, thus shaping up the following types of economic models, presented in short in graphic 2.

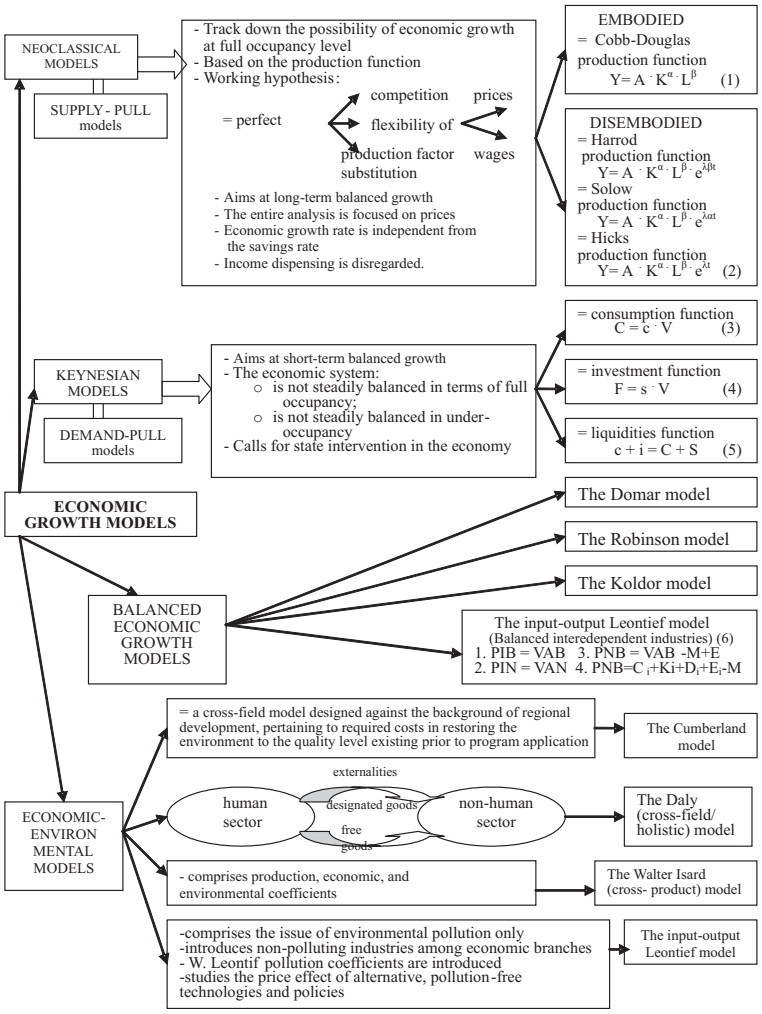


Figure 2: Tourist activity implications and interdependentions

In the meantime, improving the economic structure entails both a change in quantitative aspects such as group, branch and sector ratios, size changes, dynamic changes, and qualitative mutations covering the switch from the production diversification stage to the specialization stage, to setting up a customized profile of the national economy, to reaching a dynamic balance between various branches and activity fields, as well as switching to an optimum economic structure in a certain time span. The interdependencies and interactions between the national economy branches, between various fields and activities are enhanced, as the economy grows. Their complexity also deepens, changing from the mere exchange of materials to cooperation ties in manufacturing, to activities of technical progress extension and coverage. The increasing part played by services in the economic and social life at world level, especially for developed countries, has been interpreted by sociologists as a replacement of Şprimary civilizationŤ with Ştertiary civilizationŤ, since a service-dominated society has gradually replaced an agriculture-dominated one. As with any field of activity, the field of services has evolved from initial forms of completing primary activities to the current structures, characterized by a maximum usefulness for individuals and society alike Services have grown in modern economy, where many and various ones have emerged following societyŠs computerization, environmental protection activities or urban encroachment: household management, power and water supply, transports, telecommunications, activities targeting an increased individual leisure time (laundries, dry cleaners, modern commerce means, household appliance maintenance), as well as services related to the use of spare time (tourism, culture, sports).Being mainly made up of service-supplying, tourism represents

nowadays one of the essential components of the tertiary field.

3 The tourism as a part of the dynamic cybernetic economic system

The inclusion of tourism into the field of services stems from the way in which some of its defining traits are outlined, such as mobility, dynamism or capacity to adapt to the requirements of each tourist, as well as from the tourism product particulars, since it is the result of a balanced combination of services to which specific traits and usage mechanisms correspond. The national economy branch ledger comprises, together with other services, those encompassed by domestic and international tourism, entertainment, accommodation, transport and public food supply, thus outlining the existence of tourism as a distinct component of the tertiary field, its activities being characterized by legal aspects that are not encountered with other components of the tertiary field. Covering tourist needs, the needs of large social segments entails a whole set of processes and relationships that provide tourism with the traits of a distinctive field of activity that must be permanently adapted to the changes occurring in economic-social life, tightly correlated with other economic fields, due to the complexity and dynamism of this increasingly specific phenomenon of modern civilization. Tourist sub-products such as accommodation, transport, meals, entertainment, etc., components of the complex tourist product, are complementary to the tourism services, thus making up a part of an economyŠs services. They are the result of distinct, very well-shaped fields, which exist independently from the tourist activity, but having actual opportunities of having tourist activities inter-relating with specific ones. Tourism, as economic activity, comprises an array of services deriving from basic ones: information, tourist trip placement, accommodation, food item, alcoholic and non-alcoholic beverage retailing, treatments supplied by various spas, as well as leisure and entertainment services. Tourist demand is covered also through the balanced and complex combination of activities in other economic fields, such as: construction works, power and heating industries, car-manufacturing, electronics and electro-techniques, wood processing, fabrics, agriculture, transports, trade, communications, culture, health care, etc.

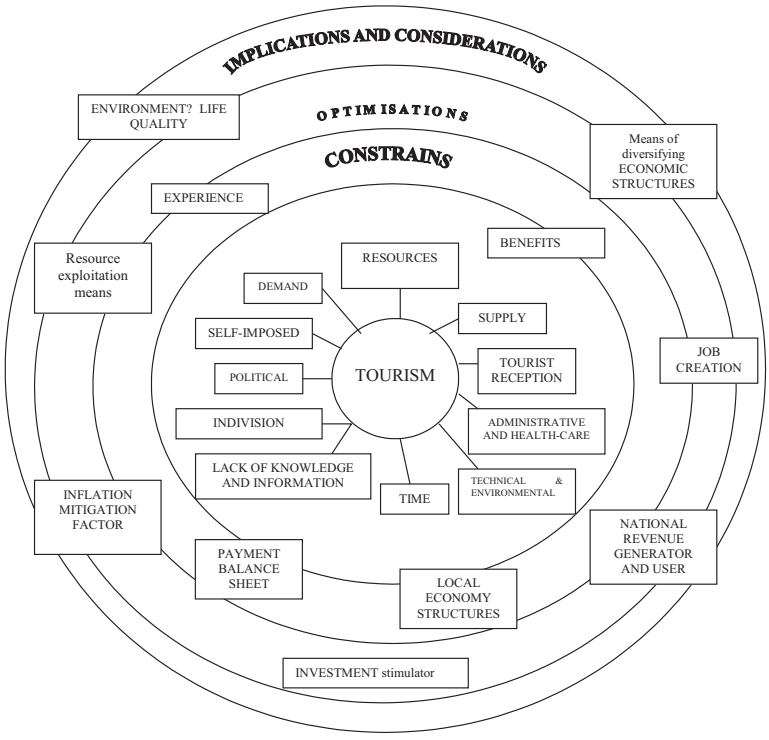


Figure 3: Tourist activity implications and interdependentions

The higher quality of elements making up the tourism product influences it in a positive way, which entails a series of corroborated actions on the others it touches all throughout tourism activity operations or on the others depending on the results of such activities. Various theories and viewpoints have been summarized in a chart (figure 3) presenting the implications and considerations of tourism activity under the influence of tourism phenomenon

constraints and optimizations on the economy and thus, implicitly, on economic growth. In systemic approach the goals (objective functions) would be:

- Benefit optimization: due to seasonality, companies are busy supplying services that are adequate to the area specifics by ma king the best of long-use facilities and investments that would bring in the best benefits;
- Local economy structure optimization: consists of stressing on the relationship between the tourism industry and local resources in order to deepen multiplying effects;
- Optimization = BOP balancing: through which companies set out to provide services to foreign guests, thus stimulating sales, assisting the economy so that outgoing tourists should be more numerous that incoming tourists.
- Experience optimization: entails making out the combination of favourite destinations and actual possibilities within the limits set by time and income; The internal and external environment generate a lot of constrains which have to be taken into consideration. The main of them may be summarized as follows :
- Tourist supply constraints: underlain by price, wealth and income;
- Attractive resource supply constraints: based on the limited resource quantitative trait;
- Technical and environmental constraints: comprise elements targeting the location-situation relationship;
- Time constraints: springing from seasonality and holiday time;
- Constraints regarding the divisibility trait of elements making up the tourist phenomenon: for example, when a tourist transportation means cannot make its rounds unless all its seats are taken;
- Administrative and health care constraints comprising drawing up administrative, customs, health care formalities that are necessary in international tourism activities;
- Political, social, ideological, conflictual and army constraints: related to the policy of every country regarding receiving tourism, domestic instability or its involvement in international conflicts, as well as natural disasters that may occur in a region;
- Reception constraints, the tourist and his comfort: comprises the existence of unfit standards for tourism reception structures, unfavorable weather conditions or the lack of natural and cultural attractions;
- Self-imposed constraints: are the result of a domestic policy adopted by a company or between companies, governmental agencies willing to develop a certain area;
- Constraints touching lack of knowledge or information: emerging due to failure to update regarding tourism activities or the activity undertaken by companies dealing with the respective tourism operation;
- Constraints related to maintenance resource limits: stemming from the capacity to valorize the financial availability of any company.

The main objectives of the cybernetic model of tourism field are the determination and the evaluations of the indicators system that must gather within a set as complete as possible and which includes regional and local indicators, sector indicators, resources indicators, results indicators and synthesis indicators.

4 Summary and Conclusions

A dynamic economy is a complex cybernetic system. The tourism is a important element of this cybernetic system continuously in interaction with the other elements of economic cybernetic system. The cybernetic model of the tourism processes should have two main goals: benefit optimization and local economy structure optimization. The cybernetic systems constraints refer to: technical and environmental constraints; political, social and ideological constraints; economic and administrative constraints; tourist-supply constraints; self-imposed constraints. The feedback of the system can be assured by the dynamic of the demand-supply ratio, the percentage of the tourist services in GIP and number of population employed in tourism activities. The model should be used to determination and evaluations of the indicators system that must gather within a set as complete as possible and which includes regional and local indicators, sector indicators, resources indicators, results indicators and synthesis indicators of the tourism activities.

References

[1] Gogonea R.M.,“Rural Tourism in Romania 1997-2005”, in Commerce Review, Nr3/2005, Bucharest, 2005.

[2] Gogonea R.M., “Statistical Methods Applied in the Analysis of the Rural Tourism”, The doctor’s degree, Academy of Economic Studies, Bucharest, 2005.

[3] Oprescu Gh., Spircu L., Zaharia M.ŞThe Basis of Economic CyberneticsŤ, Ed. Inforec, Bucharest, 1997.

[4] Zaharia M., “Services’ Economy”, Ed. Universitara, Bucharest, 2005.

[5] Zaharia M., “Quantitative methods for decision making”, Ed. ProUniversalis, Bucharest, 2004.

Marian Zaharia, Rodica Manuela Gogonea
Romanian-American University
Faculty of Domestic and International Tourist Economy
Address: 1B, Expozitiei Avenue, District 1, Bucharest
E-mail: marian.zaharia@rau.ro

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 497-503

Coalition Formation for Cooperative Information Agent-Based Systems

Nacer eddine Zarour, Sabrina Bouzidi

Abstract: The communication technology evolution led to an increase of the carried out services' and tasks' number. The aim of actual research in the cooperation and particularly the negotiation between agents is to reach a coherent global state of the multiagent system by favoring agents' synergy. In this paper, we propose a coalition formation-based negotiation model for the task allocation in the cooperative information agent-based systems. In this model, the agent that activates a negotiation seeks partners for achieving a complex task. The way that the partners take part in a coalition is done one by one according to the choice of all the coalition members. This choice is based on a multicriterion analysis. Some obtained experimental results show the suggested model performances.

Keywords: Negotiation, coalition formation, cooperative information systems, multiagent systems, task allocation.

1 Introduction

In an open environment as Cooperative Information Systems (CISs), which gathers distributed; heterogeneous; and autonomous Information Systems (ISs), the cooperation requires the intervention of a negotiation process in order to identify and structure the various problems, to propose and defend the solutions, to re-examine the intentions in a conflict, and finally to confirm the commitments [1]. In our context, each IS is modelled by an autonomous, rational, and egoistic agent.
Several negotiation models were developed like *Market mechanisms* [2], *Contract Net Protocol* [3], *Social Laws* [4], *MultiAgent Planification* [5], and *Organizational structures (coalitions, teams, congregations, etc.)* [6]. The coalition formation is a negotiation technique which is particularly preferred for solving the conflicts between agents whose behavior is economic and thus rational. A coalition is a short-term organization based on specific and contextual commitments of involving agents. This allows agents to profit from their respective competencies. To solve the task allocation problem, several coalition formation-based models were suggested [7] [8] [3]. However, these models are not adapted to the context of CISs because they do not define relations of partnership between agents, but rather relations of subcontracting. Generally in these models, the agent that activates a negotiation process will only decides to structure the coalition. It selects its partners separately according to its own desires. Consequently, the agents' autonomy is not respected.
The objective of this work is to propose a coalition formation-based model for the task allocation in CISs. The proposed model, named CFoTACIS[1] , is inspired from the reality of the cooperation between entreprises. It respects the agents' autonomy by allowing them to take part in the coalition formation process. It is recursive and includes four phases, the initialization, the negotiation, the evaluation, and the finalization.
The rest of the paper is organized as follows. In the following section, we present some related work to the coalition formation. In section 3, we develop the suggested negotiation model. Section 4 enumerates the proposed model properties. Some experimental results are presented in section 5. Section 6 summarizes the contributions of this paper and the research perspectives.

2 Related work

According to the literature, we consider two approaches for studying the coalition formation process, the macroscopic approach and the microscopic one. *The macroscopic approach*, which is based on the game theory, considers the coalition as the basic unit [9]. *The microscopic approach* considers agents as the basic unit where several models were proposed and applied in many fields, especially the e-business [10] and the task allocation [7] [8] [3]. We focuss our discussion on the models based on the second approach which deal with the task allocation problem. In [7], the authors proposed a model where agents exchange their preferences w.r.t. the possible solutions. Although, forcing agents to form alliances ensures the termination of the negotiation process, but the model does not respect agents' freedom since it imposes a solution which may not satisfy all agents.

[1]Coalition Formation for Task Allocation in Cooperative Information Systems.

In [8], the authors propose a model where the agent that activates the negotiation process first filters agents of its environment. It uses a multicriterion analysis to find agents that best satisfy its requirements. Then, it starts an argumentative negotiation with the candidates, which are classified according to its preferences. This approach does not ensure the reach of a consensus but limiting the negotiation time seems to be a reasonable strategy.
Two models are proposed in [3]. The first one is dedicated to the competitive agents. The agent selection is based on a preference model, which is built using several criteria. The major inconvenient of this model is that the representing agent of the coalition has a kind of authority on the coalition members. In the second model, agents cooperate in an altruistic way. The model ensures the reach of a consensus and has some advantages like the integration of the new partners one by one in the coalition. However, the altruistic strategy of agents does not require the gradual integration of agents in the coalition. This approach seems to be interesting in the case of egoistic agents. In [10], the authors proposed a formalization of the trust criteria which an agent grants to the other ones of the environment. This trust modeling ensures the stability in the coalition formation due to its reliability. However, the authors do not specify how to find the value of the current evaluation of trust. In order to response at this question in our proposal, we will deduce the trust from some criteria.

3 CFoTACIS: A coalition formation-based negotiation model in CISs

Before describing the suggested model, let us admit the following assumption: an agent which wants to integrate the CIS must update the portal of this CIS [4]. It must specify its localization, its know-how, the information about its branch of industry, its competencies, its capital, etc.
CFoTACIS is recursive and includes four phases, the initialization, the negotiation, the evaluation, and the finalization. In the beginning of the negotiation process, the manager which is the first coalition member, follows only the model phases. After that, when new partners join the coalition and become members, all the members try to find the next partner. The negotiation process iters until the coalition size will be reached. Let us give the essential description of the different phases

3.1 The initialization phase

The manager consults the portal of the CIS (first assumption) to obtain useful informations on agents. Then, it decomposes the global task into subtasks w.r.t. the agents' industry branches. For each subtask, the manager makes a set of the candidates among agents of the environment.

3.2 The negotiation phase

The manager simultaneously sends the proposals to agents of the same set. The message which the manager sends to the candidates must contain all information about the cooperation project, as well as the waiting period that accords to them. Then, the manager receives the responses from the candidates. Figure 1 represents the state transitions for a negotiation between a manager *(m)* and a candidate *(c)*. After receiving a proposal from a manager for cooperation, a candidate can:

- *refuse to cooperate (refuse(c, m));*
- *accept to achieve the subtask (accept(c, m));*
- *accept to achieve the subtask and also propose to achieve another subtask(s) (ok(c, m));*
- *refuse to achieve the subtask but propose to achieve another subtask(s) (no(c, m));*
- *not answer. In this case, after the expiry of a waiting period, the manager considers the not reply as a rejection (refuse(c, m)).*

If the candidate agrees to cooperate, it must send to the manager a message that contains its consideration of the cost and the time for achieving the subtask, as well as the maximum reply time that it grants to the manager to evaluate the proposal.

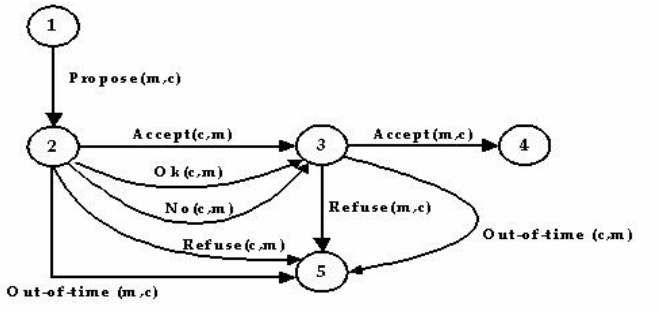


Figure 1: State transition graph of the negotiation between the manager (m) and a candidate (c)

3.3 The evaluation phase

After receiving the candidates' responses, the manager evaluates and classifies them according to their importances. The evaluation is made using a multicriterion analysis. This phase includes two stages. The first one deals with the evaluation of the criteria and the second one with the aggregation of the evaluations.

Criterion evaluation

In this stage, the manager evaluates agents which sent their proposals according to various criteria. In [11], the author defines a whole of interesting criteria for the selection of the most adequate partner. We select those which seem most significant to our context. We have classified the criteria into three categories:
(a) Criteria related to the partner
- the cooperation degree of the candidate agent with the manager (D1)
- the cooperation degree of the manager with the candidate agent (D2)
- the quality of the relation (D3) deduced from three others subcriteria:
  (i) respect of the allowed time for achieving the subtask (S1)
  (ii) the quality of achieving the subtask (S2).
  (iii) Honesty with regard to the profit distribution after the last experiment (S3)
- the experiment in the cooperation:
  (i) the cooperation number carried out by the partner (A1)
  (ii) the cooperation number carried out by the partner without interruption (A2).

(b) Criteria related to the cooperation
- the capital (C1).
- the technical capacity (C2).
- the technological competencies (C3)
- the existence of a single capability (C4).

(c) Criteria related to the candidate agent proposal The candidate must specify in its acceptance message the time T and the cost C for achieving the subtask.

The aggregation stage

After the evaluation of the different criteria, the manager incorporates the evaluations associated with all the criteria in order to have an overall estimate of each agent. In CFoTACIS, the aggregation of the criteria is used in two cases:
(i) For the trust quantification: The trust e, which results from the current relation with the agent partner, is carried out using the equation

$$e = p_1 * S_1 + p_2 * S_2 + p_3 * S_3$$

Where p_1, p_2, and p_3 are the weights that the agent must specify and $p_1 + p_2 + p_3 = 1$.
(ii) For the evaluation of each candidate: the aggregation operator is defined using a deviation at an aspiration point which gathers the preferred values of the criteria [12]. The evaluations are represented using a vector named b. The deviation b to the aspiration point a is defined by the relation.

$$deviation(a,b) = Max_{j=1..p}(\lambda_j(a_j\text{-}b_j))$$

With λ_j=1/($Ideal_j$- $AntiIdeal_j$) and *p* is the number of criteria. *Ideal* is a vector which gathers the maximal criteria values of the evaluation vector and *AntiIdeal* is a vector which gathers the minimal criteria values of the evaluation vector.
The best evaluation d* is that which minimizes the deviation to the aspiration point:

$$d^*=min\{ deviation(a,b)\}$$

For well explaining the evaluation mechanism, let us consider the following example. Agents *Ag1*, *Ag2*, and *Ag3* are evaluated according only to the criteria *criterion1*, *criterion2*, and *criterion3* in the vectors *eval1*, *eval2*, and *eval3* (table 1).

Agents	Ag1		Ag2		Ag3		Ideal	antiIdeal	λ
Criterion	eval1	λ(Ideal-eval1)	eval2	λ(Ideal-eval2)	eval3	λ(Ideal-eval3)			
Criterion1	0.18	0	0.10	1	0.14	0.50	0.18	0.10	12.5
Criterion2	0.16	0	0.12	1	0.15	0.25	0.16	0.12	25
Criterion3	0.10	1	0.18	0	0.15	0.375	018	0.10	12.5
deviation	1		1		0.50				
d*	0.50								

Table 1: An illustrative example of the adopted aggregation operator

3.4 The finalization phase

After evaluating agents' proposals, the manager classes agents in a list according to their deviations. The manager sends the agent which is at the head of the list a message for inviting it to join the coalition; else it contacts the next agent in the list, and so on.
The coalition members merge their deviation vectors to have a unified view of the candidates. This merger allows the coalition members to have supplemented information since agents have only a partial vision of their environment. The merger is carried out by the *minimum operator*. Let us consider an example showing how to make the unified evaluation vector. Agent *Ag3* is evaluated by the members *Ag1* (eval1) and *Ag2* (eval2) w.r.t. to the eleven (11) criteria (see section 3.3.1).

eval1= (0.32, 0.56, 0.88, 20, 10, 2000, 1500, 5, 1, 0.53, 0.23)

eval2= (0.56, 0.12, 0.68, 20, 10, 2000, 1500, 5, 1, 0.53, 0.23)

The unified deviation vector is

eval= (0.32, 0.12, 0.68, 20, 10, 2000, 1500, 5, 1, 0.53, 0.23)

Thus, for each subtask, the coalition members together try to choose the adequate partner.
The negotiation process will be finished when the coalition size is reached, and therefore, the coalition members start the execution of the tasks. Hence, with CFoTACIS, the decision of the coalition structure is collective.

4 The CFoTACIS Properties

The proposed model presents several advantages. The choice of the partners is carried out by the coalition members thanks to the unification process of agents' estimates carried out by each member. The adopted aggregation operator does not authorize the compensation, which permits to choose a partner which has acceptable values for all the criteria. CFoTACIS ensures the equity property because it offers the same chances to the candidates by limiting the time of the binary negotiations. When the candidate specifies its waiting time, this avoid the coalition members to send message to an agent that is no more waiting. Consequently, a considerable profit of time will be obtained. When a partner disengages, it is obvious that it will be sanctioned. This sanction is expressed by

the A2 parameter (see section 3.3.1.a). In this case, the coalition members have two alternatives for replacing the disengaged agent. They propose the subtask to the coalition members to cheek whether it exists a partner which is interested to achieve it. Therefore, a new negotiation process is avoided. The second alternative is applied when there are not members which are interested. The coalition is obliged to start a new negotiation process to find another partner. Finally, if the manager could not find partners or the required size of the coalition is not reached; the negotiation process will be remade.

5 Experimental results

We have implemented CFoTACIS using JADE 3.0 (Java Agent DEvelopment). We have compared its performances with those of a similar model [3]. We remind that in [3], the author proposed two negotiation models. We are interesting to the one dedicated to egoistic agents considering the rationality of our agents.
We have made several series of experiments. In this paper, we show the effect of varying the agents' and subtasks' number on the global time of the negotiation process (figures 2 and 3).

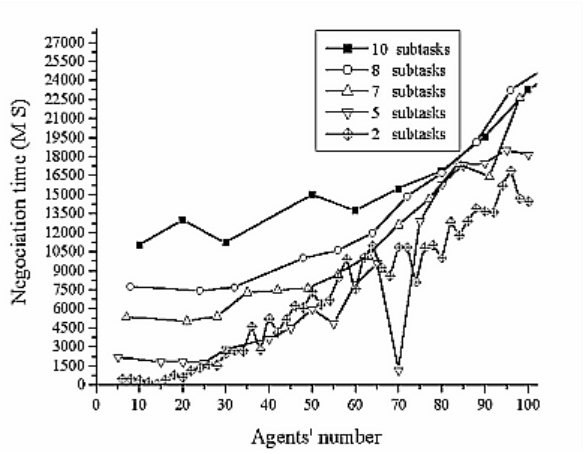


Figure 2: The negotiation time versus the agents' number for different subtasks' number in CFoTACIS

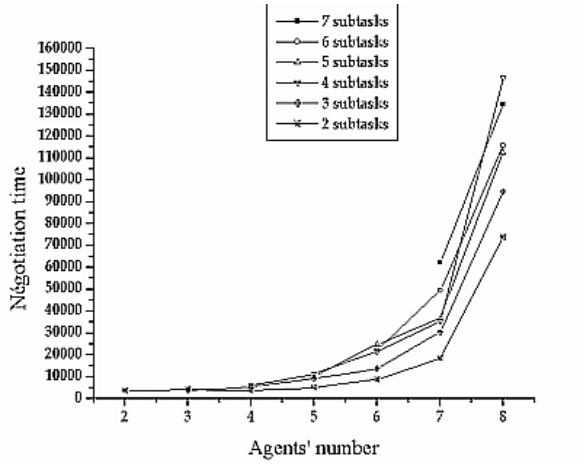


Figure 3: The negotiation time versus the agents' number for different subtasks' number in the model proposed in [3]

In each negotiation process, we fixe the subtasks' number and we increase the agents' number to finally deduce the consumed time. The graphs presented in this section are drowning using the Origin 6.0 software.

On the graph of figure 2 (CFoTACIS), we observe that we could vary agents' number from 2 to 100 and subtasks' number from 2 to 10. Whereas, on the figure 4 (the comparative model), we could vary agents' number only from 2 to 8 and subtasks' number only from 2 to 7. Therefore, we deduce that the CFoTACIS' scalability is stronger than the one of the comparative model. One reason of this result is that in the comparative model, all agents are in competition for the coalition formation. So, if we have "*n*" agents in the system, we have also "*n*" parallel negotiations which cause a scheduling problem in the system. On the other hand, we observe that the negotiation time in CFoTACIS increases as agents' and subtasks' number increase. But this increase stills reasonable and very small in comparing it with the one of the comparative model. This is due to the weak scalability of the last one. Also, in the comparative model, there is no limitation of the negotiation time between agents

6 Summary and Conclusions

In this paper, we have proposed a coalition formation-based negotiation model for the cooperative information agent-based systems, named CFoTACIS. The proposed model is inspired from the reality of the cooperation between enterprises. It includes four phases, the initialization, the negotiation, the evaluation, and the finalization. The essential properties of CFoTACIS are its consideration of the preferences of the coalition members for choosing the future partners. It does not apply any authority on agents and ensures the equity by limiting the time of the binary negotiations due to the absence of the cycles in the proposed negotiation protocol. However, it does not ensure the reach of a consensus but tries to encourage agents to cooperate. The simulation results show that in CFoTACIS the spent time for the negotiation is reasonable and the messages' number is deterministic. These results have been confirmed by comparing the CFoTACIS' performances with those of the negotiation model proposed in [3].
In the future work, we will improve CFoTACIS so that it will ensure the reach of a consensus. Also, we will adopt it in other application fields like e-commerce.

References

[1] L. Putnam, M. S. Pool, "Conflict and Negociation In Handbook of Organizational Communication: An Intersciplinary Perspective," *F.M.Jablinetal, Eds. Saga Newbury Park*,, pp. 549-599, California, 1987.

[2] T. Sandholm, "An Algorithm for Optimal Winner Determinism in Combinatorial Auctions," *In Proc. of the 16th Int. Joint Conf. on AI, Stockholm, Sweden*,, 1999.

[3] S. Aknine, "Modèles et méthodes de Coordination des Systèmes Multi Agents," *Thesis of Doctorate, University Paris IX Dauphine-UFR Science of the Organizations*,, December 2000.

[4] N. Zarour, M. Boufaida, and L. Seinturier, "A Negotiation Framework for Organizational Information Systems," *Int. Journal of information Technology and Decision Making (IJITDM). World Scientific Publishing Co*,, vol. 3, nř2, pp. 213-238, June 2004.

[5] A. E. F. Saghrouchni and S. Haddad, "Recursive Model for Distributed Plannin," *In Proc. of the 2nd Int. Conf. On Multi-Agent Systems (ICMAS'96)*,, AAAI Press. Kyoto, Japan, 1996.

[6] C. Brooks and E. Durfee, "Congregating and Market Formation," *Proc. of the 1st Int. joint conf. on Autonomous Agents and Multiagent systems*,, ACM Press, pp 96-103, 2002.

[7] A. E .F. Saghrouchni and G. Vauvert, "Formation de Coalitions pour Agents Rationnels," *In Proc. of ILIPN'2000. VIIIth days of LIPN. Multi-Agent Systemes and Formal Specification and Software Technologies* ,,Viltaneuse, France 09, 11-12, 2000.

[8] L. K. Soh, C. Tsatsouli, and H. Servey, "A Satisficing Negotiated and Learning Coalition Formation Architecture," *In V.Lesser, C.Ortiz, and M.Tambe, editors, Distributed Sensor Networks: a Multi-Agent perspective*,,Kluwer Academic Publishers, 04, 25-27, 2001.

[9] L. Larson and T. Sandholm, "Anytime Coalition Structure Generation: an Average Case Study," *Journal of Experiments and Therotical Al*,11, 1-20, 2000.

[10] Vassileva, J. Breban and S. Horsch, "Agent Reasoning Mechanism for Long-Term Coalitions Based on Decision Making and Trust," *Computational Intelligence, 18, 4, Special Issue on Agent Mediated Electronic Commerce*,, pp 583-595, Nov. 2002.

[11] C. Li and K. Sycara, "A Stable and Efficient Scheme for Task Allocation via Agent Coalition Formation," *www.2.cs.cmu.edu/ softagents/papers/CCO3.pdf*,, 2003.

[12]] N. R. Jennings, P. Faratin, P. Johnson, M. J. O'Brien, and M. Wiegand, "Using Intelligent Agents to Manage Business Processes," *Practical application of intelligent Agents and Multi-Agents technology, PAAM 96*,1999.

Nacer eddine Zarour, Sabrina Bouzidi
University Mentouri of Constantine
LIRE Laboratory, 25000 Algeria
E-mail: nasro_zarour@yahoo.fr, sab_bouzidi@yahoo.fr

Proceedings of ICCCC 2006, Băile Felix - Oradea, Romania
pp. 504-510

Support for Development and Analysis of Real Time Programmable Controller Applications

Doina Zmaranda, Gianina Gabor

Abstract: Real-time control systems have several features that distinguish them from other computerized systems. The key feature is the timely response requirement: as it is well known, for real-time systems it is important not only the functional correctness (correct results must be obtained), but also the moment when these results are obtained. This means that the timing of a real time control system must be temporally predictable, the knowledge of the timing behavior of tasks being crucial for the design of the system. In the latest years, several academic researches were carried out, but unfortunately, none of the numerous published analysis techniques and several prototype tool implementations triggers any improvements for industrial use. Rather than invent another software engineering technique, this paper describes a pragmatic approach, which fulfils clearly defined needs. The presented approach is based on a specific real-time control architecture that uses PLCs and cyclic paradigm; this is motivated by the fact that a large category of industrial applications are and could be developed using this architectural structure.

Keywords: real-time control system, event/timed triggered model, determinism, temporal behaviour

1 Real-time programming models

Real-time systems are more difficult to design due to additional constrains to be met, namely: management of physical resources, internal concurrency (multi-tasking), software distribution (which requires data and control communication) and temporal performances or safety critical behavior [8].
There is a major difference in the optimization criteria applied to non real-time and real-time systems: for non real-time applications design is oriented to obtain a good average performance; by contrast, for real-time systems the execution time must be temporally predictable, or, at least, to be easy to analyze its temporal behavior [4][13].
Another important aspect that must be considered is the way of mapping and dealing with time with respect to the resulting value and time determinism of the real-time system. A real-time system is value and time deterministic if, given a sequence of inputs, the same sequence of outputs at the same time is obtained after the process computation. A system that is value and/or time-deterministic exhibits predictable behavior (both for time and functional aspects), and thus being easy analyzable [3].
Currently, at least two real-time programming models could be considered [1]:

- **the event-triggered model** – that is based on scheduling computation. A scheduling real-time program consists of several processes that are scheduled using a specific scheduler. The necessary time for a software task to complete (task execution time) may vary depending on performance, utilization and scheduling algorithm, and is not pre-determined by the model. The execution time of each task is constrained by a deadline: the task must complete before its deadline, otherwise the real-time performance will be degraded.

- **the timed-triggered model** – is based on the idea that computation and communication are initiated at specific, predetermined time moments. In contrast to the event-triggered model, the behavior of a program that is based on the timed-triggered approach depends on program's context, not on utilization and scheduling scheme. Like in the event-triggered model, tasks are also subject to deadlines. The timed model is best suited for embedded control systems, which require strict timing predictability, than the event-triggered model. The implementation of the event-triggered model is based on scheduled processes. The design of such scheduling processes is dominated by the scheduler, in most of the situations a deadline scheduling policy being applied. This is the strength and, in the same time, the weakness of the model: because the scheduled processes are not restricted in their control flow and can be triggered and executed any time, the model is not compositional with respect to value or time determinism [8].

The implementation of a real-time system based on timed-triggered model is based on timed processes, which may run concurrently; but, as a simplifier hypothesis, a single periodic task may be considered, thus being also timed-triggered. The system designer specifies very accurate the timing for the timed processes; consequently, the

timing characteristics of the whole program is known in advance, being time safe [7]. When several processes are implied, a verification of time safety could be done (if there is enough time available for each process to execute) using execution time analysis. Checking the time safety is not easy, and requires very good knowledge about computer performance, utilization and process interactions (the context with other processes).
The best way to achieve time safety and, consequently, deterministic and predictable timing behavior is to use a timed task that is periodic. A timed task typically consists of a part that deals with process computation, and an input/output part, that deals with input/output operations from and to the memory where data on which task operates are available. Thus, the execution of the task begins with the execution of the input part (reading input data), followed by the execution of the task. Data output will be executed after the task completes, exactly at the time when the logical execution of the timed tasks elapsed. Consequently, the communication with a timed task could be realized only before and after and not during the execution of the task.
Implementation of the timed-triggered model using timed tasks leads to more precise timing than fast execution: on the other side, a value and time deterministic behavior of the system could be achieved [6].

2 PLCs and timed-triggered model

The timed-triggered task model could be implemented in real-time systems using PLCs (Programmable Logic Controllers). PLCs are a class of real-time computers mainly used for industrial control and embedded systems [14]. Because PLCs represent specialized computers used in real-time control systems, their structure and operation mode is slightly different from conventional computers, conventional computers being generally well suited to an implementation based on event-triggered model. On the other side, the hardware structure and specific operation mode made PLCs very well suited to implementations based on timed-triggered approach [2]. This assertion is proved in Figure 1 that shows the specific PLC program cycle. According to the program structure from Figure 1, the PLC program can be viewed as a single timed task that executes periodically. It has an input and an output part, as it was described in the timed model; besides, some parts regarding communication and processor overheads were added.

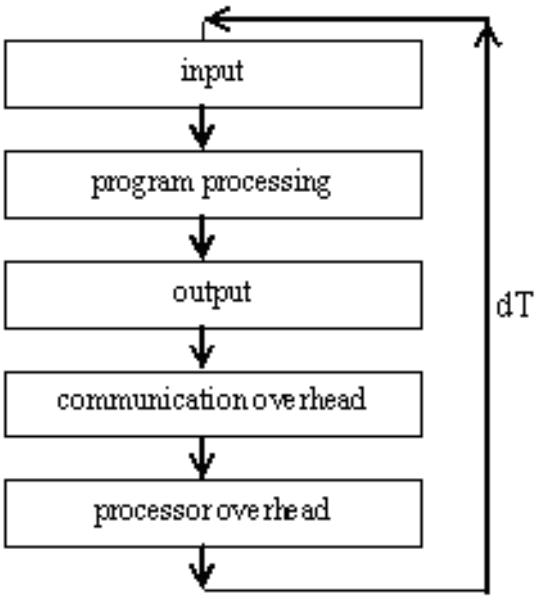


Figure 1: Program structure for PLCs

Estimation of time behaviour for a PLC program relies on estimation of maximum scan time of the control program. The diagram below shows the main operating cycle time steps for PLCs. These are the following [9]:

- input – reads the status of input modules and updates the input image in the processor with this information (read into the data table from processor's memory)
- program scan – executes the program processing, based on the inputs

- output scan –updates the output image and transfers the output image information to the output modules
- communications – takes place communication with programmers and other network devices
- processor overhead – takes place several processor internal housekeeping actions: performing program pre-scan, updating the internal status files

The development of a PLC application usually requires [12]:

- configuration of PLC architecture
- selection of number, type and address of the inputs and outputs used
- writing and debugging the control program

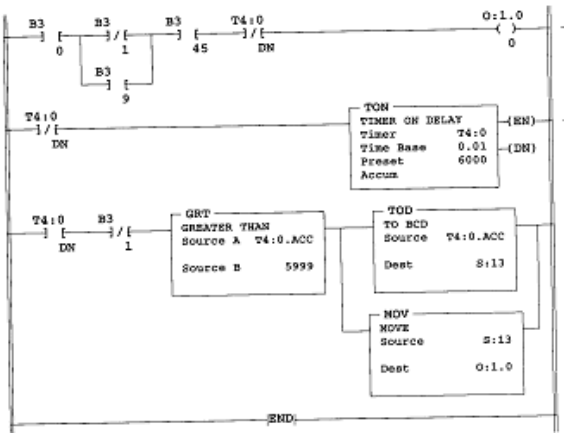


Figure 2: An example of ladder program

The main issues when analyzing the time safety for a PLC program implementation is to assess correctly the time base (dT). The time base depends of processor speed, the length of the control program, as well as the number of input/outputs connected.
All above factors have to be taking into account when estimating dT. Calculations should be done based on the actual configuration of the system (number and type of input/output/modules, communication issues, program instructions on several possible paths of the control program) and on specific values given by the producers. The evaluation of program (ladder) execution time is based on the execution times for each type of instruction (assuming the true state) given by the producer; these values must be added for each rung.
For example, for the very simple example program form Figure 2, the total estimated execution time was the following:

- first rung: 38 microseconds
- second rung: 139 microseconds
- third rung: 288 microseconds

Results a total of 465 microseconds estimated time. This time should be used further for calculating the overall execution time that includes all the factors that were listed above. An important fact that must be considered is that, usually, a control program could execute different program paths (when possible, implemented with subroutines) depending on the input data values. Estimation of execution time should consider all possible paths, and rely on the maximum calculated value.
The Table 1 describes the main steps in estimating minimum and maximum execution time for a PLC program, using an example for Allen Bradley SLC-500/04 processor [15]. All values are given in microseconds. Other types of analog input and output modules are not considered in the above table. If there are present, they should be added with specific number/specific time value.

>From the above example, we can conclude that scan time analysis provides two estimated values: for minimum and maximum estimated time. Because, according to the PLC program execution mode, it is possible that an input change to be observed only in the second program execution cycle, in order to assure time safety the following relation should be accomplished:

$$T_{deadline} \ll 2\, t_{execution\_time\_max}$$

Because, $t_{scan\_time\_max}$ is considered the worst case for hard real-time requirements, although from soft real-time requirements, the minimum value could be also considered. If the above assertion holds, in the given execution environment the performance is faster than required by a particular application under worst case conditions. For this case, the hardware is "over dimensioned" and performance concerns are not an issue.
If imposed deadline is as the same order of magnitude as 2 $t_{execution\_time\_max}$, than careful testing of temporal properties of implemented code is required. Parts with tight deadlines should be inspected in order to avoid situations, like when program code flows against the program scan [11],[9].

1. Estimate the input scan time for input modules				**Minimum**	**Maximum**
Type of module	Number	Specific value	Total	= total input time +101	=minimum scan time + number of analogue x 50
Discrete 8 point	2	197	394		
Discrete 16 point	1	313	313		
Discrete 32 points	0	545	0		
Analog combo input	1	652	652		
Total input time			1359	**1460**	**1510**
2. Estimate the output scan time for output modules					
Type of module	Number	Specific value	Total	= total input time +138	=minimum scan time + number of analog x 50
Discrete 8 point	1	173	173		
Discrete 16 point	3	272	816		
Discrete 32 points	0	470	0		
Analog combo output	1	620	620		
Total input time			1609	**1747**	**1797**
3. Estimate the program execution time					
Number of rungs	3	1	3		
Program execution time	all instructions are true		465	**468**	**468**
4. Processor overhead time				**178**	**278**
Total input + output + program+overhead				**3853**	**4053**
5. Communication overhead					
Background	total x 1(min) ; total x 1.4 (max)			3853	5674
Foreground	+ 0 (min) ; +2310 (max)			3853	7984
Total execution time	**divided by 1000 for ms**			**3.853 ms**	**7.984 ms**

Testing the actual PLC program and measurement-based evaluation of execution time could be also done, if supported, using several PLC's facilities:

- *processor's test mode*: the ladder execution program begins, but the states of outputs are not written to the real I/O devices; it is used mainly for testing program functionality
- *processor single-scan mode*: the ladder execution begins, but only for one scan

Some PLC processors also provide status bits that contain values for current scan time and/or average scan time; measurement-based assessment of program scan time could be also done using these values [5]. But it is important to notice that such values consider only the actual program and doesn't take into account the whole possible program paths that could lead to different execution time values.

3 Support tool implementation

In the literature, it is often claimed that time analysis and estimation tools for real-time systems should support generic software structures. But, this requirement makes analysis and development of such tools very complex. It would be a better strategy that, in development of the analysis tool, to exploit simplifications from the concrete application context [10]. As a potential benefit, the precision of the WCET analysis tool will be improved and also the implementation complexity will be reduced.
Starting from the above idea, a support application was developed, dedicated to timing evaluation for the specific domain of real-time embedded and control systems that were implemented using PLCs. The application was written in VisualC and permits to set all the characteristics for a given controller. It permits that all calculations to be done automatically, according to the procedure given in the above example.
However, to obtain program execution time estimation, requires knowing and testing every relevant execution scenarios of the code. A concrete execution scenario is determined both for the initial state and for the values of actual input parameters. Because of the linear structure and cyclic operation of the PLCs program code, an exhaustive analysis of all program paths is, in this case, feasible. In order to evaluate minimum and maximum execution program times, calculations are based on information regarding program structure that is described using a graph structure. Consequently, at the beginning of the time analysis, a model of the application, based on application structure is generated, using a weighted graph. The graph abstracts the program structure indicating all possible paths in the program code using several nodes; transition to each node is annotated with information regarding the number of rungs/path. It is important to construct an accurate execution graph model, in order to obtain an accurate evaluation. When the complexity of the model increases too much, safe approximation should be used. An example for a given program structure is presented in Figure 3.
In the above example, according to the PLC program structure, it was supposed that the program consists of a main part and several subroutines that were called or not depending on initial conditions. If some subroutines are disjunctive for a set of input data, they are represented in parallel; otherwise, a serial approach is used. The values for each edge were established according to the length of the path (time/path –because each subroutine has only a single-path, time execution could be easily estimated). The return paths are denoted with value 0, in order not to affect the overall total time calculations.
Based on the constructed model, and using maximum graph flow algorithms, the maximum path could be determined for the given graph, and, for this the maximum execution time could be calculated [16]. Maximum path is chosen based on the idea that, given a set of initial conditions, all possible paths are considered. The calculated value will be then considered further in the scan time estimation part of the application, for calculating the total scan time.
The framework makes all calculations according to the principles given in Table 1, and based on execution program time given by the previous graph estimation. Using this framework one can assess the maximum scan time for a given type of controller. For the time being, the number and type of instruction are given to this framework manually; for the future, it is intended to develop a tool which parses the program file and provides to the framework automatically the number and type of program instructions for further calculations.
The simplified program structure of code programs in the controller makes timing analysis easier. Being an application specific framework specifically designed for such controller-in-the-loop implementation, its limitations are evident; one of the most important one results from the fact that the tool provides a static evaluation method. Consequently, it produces safe upper bounds, but with the price of overestimation used during the calculations. But, hard real-time applications require a provision of safe upper bounds, so the developed application proves its practical feasibility.

4 Conclusions

This paper presents some real-time system models, focusing on the timed-triggered model considered the support for the development of value and time deterministic real-time programs that are applied in the context of embedded and control systems. A possible software architecture that can be used is described; the way these enhancements help the programmer to emulate the dynamic operation of a specific application to alternative scenarios being also presented. For this one, a software tool was developed, tool that could be used for timing properties estimation of real-time control systems that are implemented using PLCs. Using this tool, the way of estimating the control program execution time is investigated; a practical method, motivated by the observation of every-day real-time control systems, that uses time to achieve predictability of activities, is developed. In this idea, common sense

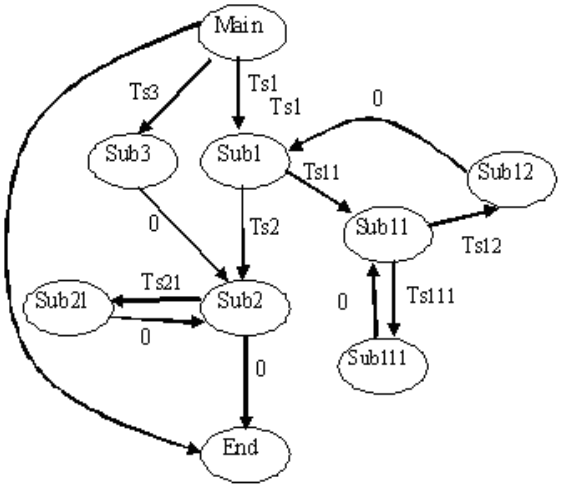


Figure 3: Graphical modeling of a program structure

concepts of time provide means for orientation, regulation and coordination; this practical significance can guide the development and analysis of real time control systems. A brief example that illustrates the analyse steps of a particular controller application is described. These steps were integrated into an application that automatically estimates the timing properties for a given program.

References

[1] Kirch, C.: *Principles of Real-Time Programming*, EMSOFT02, LNCS 2491, Springer-Verlag Berlin, 2002

[2] Svrcek W., Mahoney D., Young B.: *A Real-Time Approach to Process Control*, John Wiley & Sons, 2000

[3] Laplante Ph. A. : *Real-Time Systems Design and Analysis – An Engineer's Handbook – Second Edition*, IEEE Computer Society Press, 2000

[4] Kopetz, H.: *Real-Time Systems - Design Principles for Distributed and Embedded Applications*, Kluwer Academic Publishers, 1997

[5] Rullan, A.: *Programmable Logic Controllers versus Personal Computers for Process Control*, Computers industrial Engineering, vol 33, no.1-2, 1997

[6] Irwin G.W.: *The Engineering of Complex Real Time Computer Control*, Kluwer Academic Publisher, 1996

[7] Barbat B., Filip F. Gh. : *Ingineria programării în timp real*, Editura Tehnică Bucureşti, 1997

[8] Krishna, C. M., Shin K. G.: *Real-Time Systems*, McGraw-Hill Companies Inc., 1997

[9] Smith C.A., Corripio A.B.: *Principles and Practice of Automatic Process Control, 2nd edition*, John Wiley & Sons, New York, 1997.

[10] Richard, P.: *A Tool Controlling Respons Time of Real Time Systems*, LNCS 2324, Springer-Verlag Berlin, 2002

[11] Shinskey F.G. : *Process Control Systems: Application, Design and Tuning, 4th edition*, McGraw Hill, New York, 1996

[12] Hohmann, T.: *Why PCs Won't Kill PLCs*, Industrial Computing, vol. 15, no. 10,1996

[13] Kavi, K. M., Yang, S. M.: *Real-Time Systems Design Methodologies: An Introduction and a Survey*, Journal of Systems and Software, 1995

[14] Levine P. S.: *The Programmable (Logic) Controller: Adapting in an Environment of Change*, Industrial Computing, vol. 14, no. 3, 1995

[15] Rockwell Automation : Allen Bradley Automation Systems, 1995.

[16] Cretu, V.: *Structuri de date si tehn ici de programare*, vol. II, Editura Orizonturi Universitare, Timisoara, 2000

Doina Zmaranda, Gianina Gabor
University of Oradea
Department of Computer Science
Address: 1, Universitatii Street, 410087 Oradea
E-mail: {zdoina,gianina}@uoradea.ro

Author index