INT J COMPUT COMMUN, ISSN 1841-9836
7(5):968-975, December, 2012.

The Specification of ETL Transformation Operations based on Weaving Models

M. Vučković, M. Petrović, N. Turajlić, M. Stanojević

Milica Vučković, Marko Petrović,
Nina Turajlić, Milan Stanojević
University of Belgrade, Faculty of Organizational Sciences
Serbia, 11000 Belgrade, Jove Ilića 154
E-mail: {milica.vuckovic, marko.petrovic,
nina.turajlic, milan.stanojevic}@fon.bg.ac.rs

Abstract: In the ETL process the transformation of data is achieved through the execution of a set of transformation operations. The realization of this process (the order in which the transformation operations must be executed) should be preceded by a specification of the transformation process at a higher level of abstraction. The specification is given through mappings representing abstract operations specific to the transformation process. These mappings are defined through weaving models and metamodels. A generated weaving metamodel (GWMM) is proposed giving the complete mapping semantics through specific link types (representing the abstract operations) and appropriate OCL constraints. Weaving models specifying the actual mappings must be in accordance with this proposed GWMM.

Keywords: ETL process, MDD, Weaving models.

1 Introduction

In crisis management the tracking of a large amount of information (regarding people, material, financial, medical and other resources) is crucial. The establishment of a data warehouse, in which all of the relevant information could be easily stored, processed and analyzed, would enable the crisis management coordinators to make efficient decisions.

One of the most demanding phases in the data warehouse design process is the design of the process for transforming the source data into a form suitable for its further analysis (the Extract-Transform-Load process). Mistakes made during this phase may lead the whole project to failure. Since it is usually very complex and time-consuming it is necessary to provide data warehouse designers with adequate techniques to aid them in overcoming this complexity.

The ETL process consists of the Extract, Transform and Load phases. The focus of this paper is the Transform phase of the ETL process. This transformation process involves the execution of a set of operations through which the actual transformations are achieved. Most of the existing approaches directly define the order in which the transformation operations must be executed during the transformation process. We consider this to be too complex because it involves the definition of the process realization at a low level of abstraction and propose that this phase should be preceded by a specification of the transformation process at a higher level of abstraction. The main focus of this paper is the specification of the key abstract operations specific to the transformation process. These abstract transformation (AT) operations denote the semantics related to the different possible types of correspondences that exist between the source models and the target model and are the basis for the specification of mappings.

We propose an approach in accordance with Model Driven Development (MDD) which is based on the premise that the most important product of software development is not the source code itself but rather the models representing knowledge about the system that is being developed. The main goal of MDD is to automate software development through the successive

application of model transformations, starting from the model representing the specification of the system and ending in a model representing the detailed description of the physical realization, from which the executable code can ultimately be generated.

In accordance with MDD a special kind of model i.e. a weaving model (WM) is used for the specification of mappings between heterogeneous models [1–3]. Through these weaving models the correspondences between individual elements of different models (called woven models) are defined. In compliance with the OMG MDA an appropriate metamodel (i.e. a weaving-metamodel WMM) is defined and weaving models must conform to it. The WMM actually defines the types of correspondences which may occur between particular concepts of concrete models. However, metamodel concepts (e.g. relational metamodel concepts) cannot be used in the definition of the WMM, hence only new link types can be defined. This implies that the corresponding WM i.e. the mappings between concepts of concrete models (e.g. a concrete relational model) cannot be semantically controlled.

In this paper mapping models and metamodels are used for specifying the AT operations. In the proposed solution we provide for the explicit introduction of metamodels into the weaving model approach as well as an appropriate metamodel for the semantic mapping between these metamodels (the generated WMM or GWMM). The AT operations are represented through the concepts of this GWMM. These concepts allow the establishing of semantic correspondences only between those metamodel concepts on which the respective abstract operations can be applied. Since weaving models must conform to the GWMM this implies that both the syntax and the semantics of the correspondences between concepts of concrete models can be controlled.

The paper is organized as follows. The next section describes the main issues of the design of the ETL processes and briefly presents the related work regarding the different approaches to its design. Section 3 describes the existing MDD approach used in the Eclipse Modeling Framework for the specification of model mappings. The proposed solution for the specification of the GWMM as well as several examples demonstrating its application, are given in Section 4. Finally, a conclusion is given detailing the main benefits of the proposed approach.

2 Design Issues

One of the main issues in data warehouse is the problem of data integration, i.e. the integration of heterogeneous data sources into a single source. To this end first a global model i.e. a reconciled model should be created giving a unified view of the relevant source data. The main benefit of this approach is that it creates a common reference data model for the whole organization [7]. After the reconciled model has been created the next step is the population of the data warehouse (DW) with the actual data. If the chosen architecture for DW design presumes that the reconciled model is materialized, first the reconciled data layer will be populated from the sources and then the DW we be populated from the reconciled data layer. We have adopted such an approach in this paper since it enables the clear separation of source data extraction and integration from DW population [7]. Therefore, the data extracted from the sources first needs to be transformed into a format compliant with the defined reconciled model.

Assuming that a single reconciled model has previously been created on the basis of the source models (e.g. by overlapping the source models) we propose that the first step in the transformation process design should be the specification of the correspondences between the source model concepts and the reconciled model concepts at the highest level of abstraction. At this stage conflicts can occur both at the structure and the instance levels (i.e. the data level). Structure level conflicts are caused by the fact that different structures, and relationships among them, are used to represent the same real world concepts in different source models. Instance level conflicts are more complex and may be the result of different granularity levels (e.g. events

recorded on a daily or weekly level) or different formats in which the data is recorded.

At the structure level we can differentiate between mappings at two different granularity levels i.e. the element and attribute level. Element level mappings are used to define the correspondences between source model and reconciled model concepts by which same real world concepts are modeled. These mappings represent different abstract transformation (AT) operations (e.g. *Join*, *Equivalence*, etc.) by which the semantics of the correspondences between the related elements are defined and which are easily understandable from the end-user viewpoint. The AT operations are not executable and will be transformed, in the subsequent phases of the ETL process design, into one or more actual operations (e.g. *SQL JOIN*, *UNION*, etc.), though this transformation is not in the scope of this paper.The attribute level mappings are at a lower granularity level and give the details of the established element level mappings. Attribute mappings represent AT operations that transform the values of one or more source model attributes into one or more reconciled model attribute values (e.g. *Equals*, *Concatenate*, *Split*, *Add*, etc.).

Most of the existing approaches to ETL design proposed in literature or implemented in commercial tools do not provide concepts which allow the explicit and formal definition of the semantics of the element or attribute mappings. In [11] mechanisms for the specification of the most common ETL process operations (e.g. *Join*, *Filter*, etc.) are provided and a set of corresponding UML stereotypes is defined. These mechanisms are related through UML dependencies, and attribute mappings are defined by notes attached to the dependencies. *Data mapping diagrams* based on the Data Mapping UML Profile are introduced in [10] to trace the flow of data and are organized into four levels (*Database*, *Dataflow*, *Table* and *Attribute levels*) through the use of UML packages. At the table level only data relationships are specified and not the actual processes therefore, these mappings do not carry any semantics. At the attribute level the semantics of the mappings are given either as UML notes attached to the target attributes if the relationship is represented as an association, or, if the relationship is represented as a mapping object, by the tag definition of the mapping object. In [12] a static conceptual model of the ETL process is proposed for identifying the transformations in the ETL process and it includes the transformation entity (an abstraction of modules of code executing a single task related to filtering, cleaning or transformation operations), ETL constraints (regarding the necessary data requirements), notes (explaining the semantics of the applied functions: the type or expression/condition). In these approaches most of the transformations semantics are given through notes, often in a natural language, which does not represent a formal specification.

On the other hand, most approaches directly define the order in which the transformation operations must be executed during the transformation process i.e. the dataflow, as in [10, 11]. Also in [9] ETL process models are designed in accordance with the introduced BPMN4ETL metamodel (based on the Business Process Modeling Notation) which defines the dataflow. We consider that the specification of the transformation process realization should be preceded by a specification of the process at a higher level of abstraction. These specifications should be formal enough to enable the designer to move to lower specification levels resulting in the dataflow specification, through certain model transformations (in accordance with MDD). To this end, in [12] a static conceptual model of the ETL process is proposed which can be mapped into a logical model representing the workflow. In [8] a UML profile is proposed which introduces the *MappingOperator* stereotype (among others) to tag classes defining the functions that can be used for specifying the mapping expression of a particular mapping.

Therefore, we propose that the specification of the dynamics of the ETL process should be preceded by a static specification in which the mappings are defined through concepts which explicitly represent the semantics of the transformation operations. This specification should be formal enough to allow its transformation into the dataflow specification (in accordance with MDD), it should be extensible to allow the designers to add their own concepts, and its concepts

should be easily understandable from the end-user viewpoint.

3 The AMW Approach

In accordance with the MDD motto that everything should be treated as models, the weaving model approach treats mappings between heterogeneous models also as models [1, 2] and is supported by the ATLAS Model Weaver (AMW) toolkit [3] within the Eclipse Modeling Framework (EMF) environment. In the AMW approach weaving models (WM) are used for defining the correspondences between individual concepts of different models (called woven models).

In the context of the AMW approach mapping specifications can be regarded at different abstraction levels (illustrated in Figure 1.), which actually correspond to the different levels in the OMG MDA standard [4]. Taking into account the relationships which must exist between models from different abstraction levels in the OMG MDA, a mapping model at a given level of abstraction serves as a metamodel for mapping models from the lower abstraction level. Or, in other words, every mapping model must conform to a mapping metamodel from the higher abstraction level. It should be noted that in this paper we concentrate only on the metamodel (M2) and model (M1) levels.

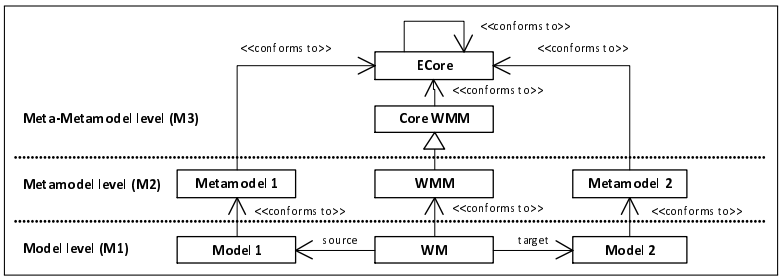

Figure 1: The AMW Approach

In accordance with the previous statements, a WM must also conform to a particular weaving metamodel (WMM). This WMM is actually an extension of the generic WMM i.e. the Core WMM, which is based on the meta-metamodel namely, the ECore meta-metamodel in the EMF environment. Since the concepts of the Core WMM do not give the definition of the specific semantics of the correspondences, it should be extended [1] to provide the semantics relevant for the specific context in which the models are used (i.e. it should be extended to include concepts specific to a certain domain). Therefore, the WMM will actually define the types of correspondences which may exist between concepts of woven models, and consequently correspondences specified in a WM must only be instances of the types defined in the WMM.

However, a WMM cannot specify the mappings between metamodel concepts (e.g. the relational metamodel, the XSD metamodel concepts etc.), it merely defines new correspondence types. More precisely, the semantics of an introduced correspondence type are given only through the definition of its name without specifying the type of metamodel concepts that may participate in that type of correspondence. Thus, one of the main drawbacks of the AMW approach is the fact that a WMM does not include the necessary semantic rules for establishing mappings between metamodels so mappings between meta-concepts cannot be defined. Instead it is only possible to define mappings between concepts of concrete models. Consequently, only the syntax of these mappings can be controlled and not their semantics. In other words, it is up to the designer to know these semantic rules and ensure they are fulfilled when defining the mappings.

In [5,6] a solution is proposed for overcoming this problem in the context of the specification of mappings between heterogeneous schemas. This solution is based on the explicit introduction of

meta schemas into the WM approach as well as an appropriate WMM for the semantic mapping between the concepts of these meta schemas. The details of this approach and it application for the specification of AT operations are given in the following section.

4 The Proposed Solution

To take into account the assertions made in the previous section an extension of the Core WMM is proposed which includes the concepts necessary for providing the semantics of the correspondences in the context of the transformation process. In Figure 2. this extended WMM is shown using the UML class diagram. New mapping link types representing the identified AT operations (*Join*, *Equivalence*, *Equals*, *Concatenate*, etc.) are defined as specializations of the WLink concept. It should be emphasized that every identified AT operation is represented by a separate *MappingLink* type in our extended WMM though we have only depicted those relevant for the following examples. These new *MappingLink* types are used for the specification of mapping between the concepts of different metamodels (the actual concepts are represented by the *MappingLinkEnd* subclass of the *WLinkEnd* class). References *source* and *target* enable the defining of many-to-many mappings between the concepts of different metamodels. *MappingModelRef* and *MappingElementRef* represent references to concrete UML models and their elements.

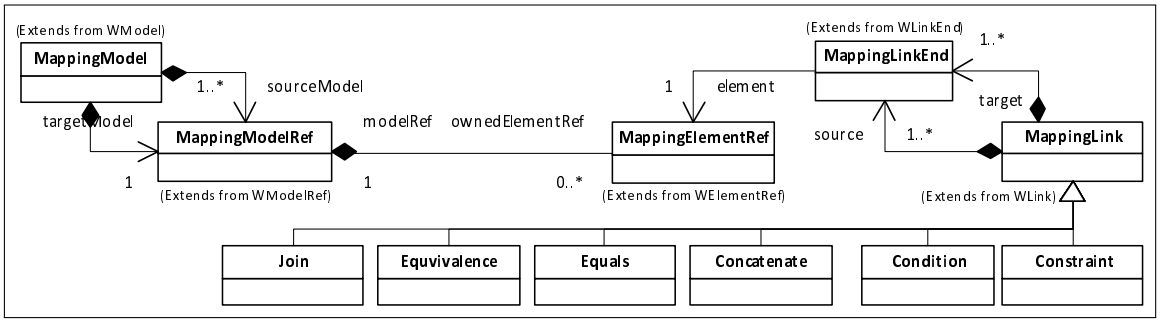

Figure 2: Extended Weaving Metamodel (simplified)

As explained in the previous section the extended WMM merely defines new mapping link types (whose semantics are only given through their names). Since each specific type of mapping assumes certain constraints regarding the number and type of model concepts that may participate in that type of mapping (e.g. the *Join* operation may only be used for mapping elements and must map at least two source model elements to a single reconciled model element) these constraints should be included in a WMM to prevent structural and semantic mistakes in WMs.

The procedure for generating the GWMM is based on the transformation given in [5, 6]. The transformation is accomplished by defining an individual WM for the mappings between metamodel concepts (the MM-WM) and then transforming it into a generated WMM with appropriate OCL constraints (e.g. for checking whether the type of model concepts involved in each mapping are valid). Specifically, a WM conforming to the proposed extended WMM is created in which the mappings are defined through a set of mapping links which are instances of the introduced mapping link types (e.g. *TableJoin* is an instance of the *Join* link type). This WM is then transformed into a new generated WMM (GWMM) in which the semantics of these mapping links are represented by appropriate OCL constraints. This GWMM now serves as a metamodel for weaving models at a lower level of abstraction (the M1 level in the OMG MDA). More precisely put, while the extended WMM defines the new mapping link types (representing the AT operations), it is the GWMM with OCL constraints that gives the complete semantics

of the defined mappings (regarding the metamodel concepts on which those operations can be applied). This process is illustrated in Figure 3. It is assumed that all of the models (both source and reconciled) are expressed using the same formalism i.e. they are based on the same metamodel, which is common practice in DW design. For the purpose of this paper we have chosen the relational metamodel. The concrete models are represented as UML class diagrams using the appropriate UML stereotypes defined by the relational metamodel (the definition of these UML stereotypes is omitted from this paper due to space constraints).

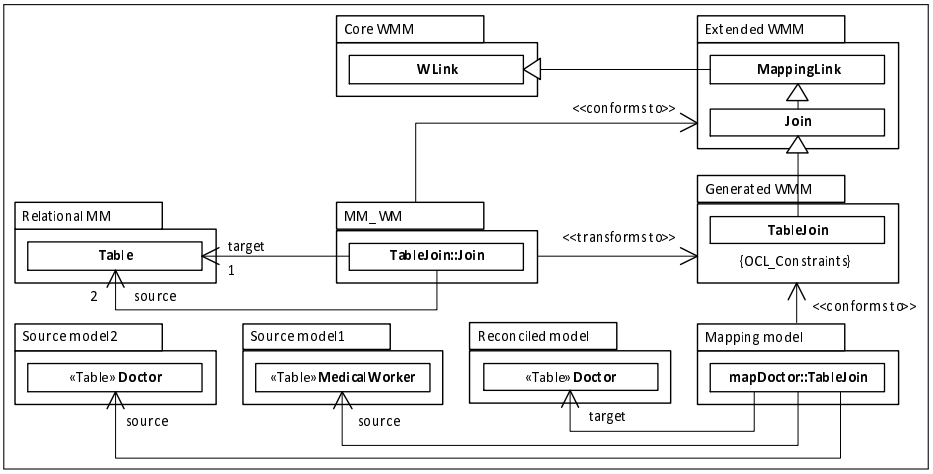

Figure 3: Proposed Solution

Next we give several examples of concrete weaving models conforming to the proposed GWMM. It should be emphasized that the first step in creating a particular WM is to define the element mappings. Subsequently, the attribute mappings are defined detailing the established element mappings. All mappings are defined by using the concepts of the proposed GWMM.

In Figure 4. an example is given illustrating the abstract *TableEquivalence* operation. The *MapDoctor* mapping is an instance of the *TableEquivalence* mapping link which states that the mapped elements *MedicalWorker* and *Doctor* are equivalent in the sense that they describe the same real world concept at the same abstraction level. The details of the *mapDoctor* element mapping are given through the child attribute mappings. The attribute mapping *mapFullName* is given by the *ColumnConcatenate* mapping link whose semantic indicates that the values of the *FirstName* and *LastName* attributes of the *MedicalWorker* element should be concatenated to obtain the value for the *FullName* attribute of the corresponding *Doctor* element. The *mapSSN* mapping is given by the *ColumnEquals* mapping link whose semantic indicates that attributes *SSN* of the *MedicalWorker* and *Doctor* elements exactly coincide.

Another example is given in Figure 4. (on the right) illustrating the situation in which the *MedicalWorker* element of the fist source model and the *Doctor* element of the second source model represent the same real world concept but record different information about it. Therefore they are represented by a single *Doctor* element in the reconciled model which includes all of the relevant information contained in both of the source models. For defining this type of mapping the abstract *TableJoin* operation is used whose semantic indicates that the *MedicalWorker* and *Doctor* elements of the two different source models should be joined to obtain all of the relevant information for the *Doctor* element in the reconciled model (the *mapDoctor* mapping link). In addition to defining the corresponding child attribute mappings it is also necessary to define the condition by which the elements should be joined. This is given by the *ColumnCondition* mapping link, also a child of the defined element mapping, which specifies that the join should be performed on the basis of the *SSN* attributes. The *ColumnConstraint* link, again, also a

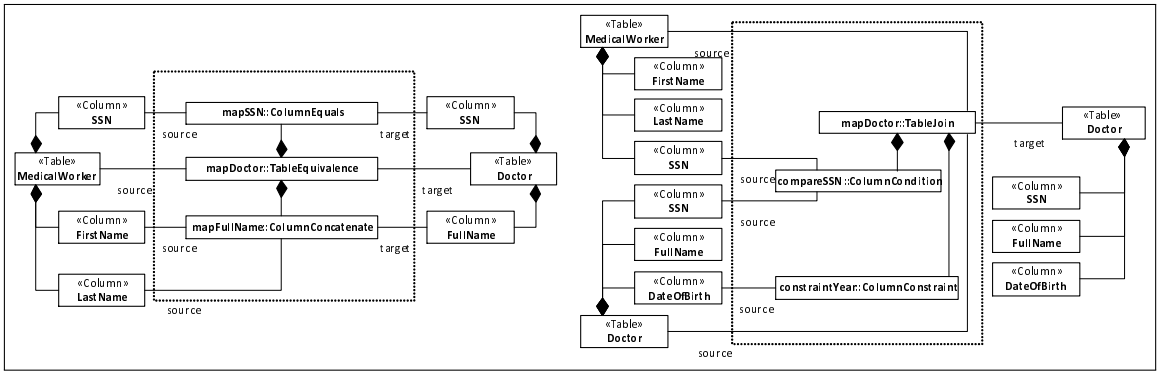

Figure 4: Illustrations of the proposed approach

child of the defined element mapping indicates that only those instances of the *Doctor* element which satisfy the defined *constraintYear* regarding the required *DateOfBirth* will be included in the transformation. The child attribute mappings giving the details of the *mapDoctor* element mapping have been omitted from because they would render the picture unnecessarily complex.

5 Conclusion

To overcome the compexity of ETL process design they should be designed gradually through the development of a series of models and the corresponding transformations between them (in accordance with MDD). The aim of this paper is to facilitate the defining of the transformation process at the highest level of abstraction. The specification of the transformation is given through mappings between the source models and the reconciled model which represent abstract operations specific to the transformation process. In the proposed solution the specification of the AT operations is based on the weaving model approch. To this end, first a description of the existing model weaving approach in the EMF environment is given and its drawbacks are discussed. Then, a solution is proposed which actually extends the existing AMW approach to overcome the identified drawbacks. The proposed solution is based on the introduction of a special generated WMM with OCL constraints through which the mapping links between concepts of concrete models are semanticaly controlled. Namely, the AT operations of the ETL process are actualy represented through appropriate semantic mapping links between metamodel concepts.Therefore, the main benefit of the proposed approach is the introduction of a formal specification of the semantics of the transformation operations giving a static view of the transformation process which is extensible and easily understandable from the end-user viewpoint. It is also platform independent so it can be used to complement other approaches dealing with the dynamic aspects of the transformation.

On the basis of these specifications the designer would, through certain model transformations (in accordance with MDD) move to lower specification levels which would result in the data flow specification i.e. the specification of the process dynamics (the AT operations would be transformed into one or more actual operations e.g. *SQL JOIN*, *UNION*, etc.). Therefore, further work in this area would be aimed at the realization of the proposed solution in the EMF environment. The specification of the AT operations given through weaving models would be used for obtaining the transformation models in a given transformation language (ATL, QVT or XSLT). These transformation models can be used for the automatic generation of code for any target platform.

Acknowledgment

The research presented in this paper was partially supported by the Ministry of Education and Science of the Republic of Serbia, Grant III-44010 and TR32013.

Bibliography

[1] Del Fabro, M.D., Bézivin, J., Jouault, F., Valduriez, P., *Applying Generic Model Management to Data Mapping*. In proc. of Base de Donnes Avances (BDA 2005), France, 2005.

[2] Del Fabro, M.D., Valduriez, P., *Semi-automatic model integration using matching transformations and weaving models*. In proc. of symposium on Applied computing, ACM, 2007.

[3] Del Fabro, M.D., Bézivin, J, and Valduriez, P., *Weaving Models with the Eclipse AMW plugin*. In: Eclipse Modeling Symposium, Eclipse Summit Europe 2006, Germany, 2006.

[4] Miller, J., Mukerji, J., *Model Driven Architecture (MDA)*. http://www.omg.org; 2001.

[5] Nešković, S., Vučković, M., Aničić, N., *On Using Weaving Models to Specify Schema Mappings*. 2nd Intl Workshop on Future Trends of Model-Driven Development, Portugal, 2010.

[6] Aničić, N., Nešković, S., Vučković, M., Cvetković, R., *Specification of Data Schema Mappings using Weaving Models*. Paper accepted for publication in ComSiS Journal, March 2012.

[7] Golfarelli M., Rizzi, S., *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill, ISBN-13: 978-0071610391, 2009.

[8] Kurz, S., Guppenberger, M., Freitag, B., *A UML profile for modeling schema mappings*. In Proc. of the intl. conf. on Advances in Conceptual Modeling, (pp. 53–62), Springer, 2006.

[9] El Akkaoui, Z., Zimányi, E., Mazón, J.-N., Trujillo, J., *A model-driven framework for ETL process development*. In Proc. of 14th intl. workshop on DW and OLAP, ACM, 2011.

[10] Lujan-Mora, S., Vassiliadis, P., Trujillo, J., *Data Mapping Diagrams for Data Warehouse Design with UML*. In Proc. 23rd Intl. Conf. on Conceptual Modeling, Springer, 2004.

[11] Trujillo, J., Luján-Mora, S., *A UML Based Approach for Modeling ETL Processes in Data Warehouses*. Conceptual Modeling - ER 2003, (pp. 307–320), Springer-Verlag, 2003.

[12] Simitsis, A., *Mapping conceptual to logical models for ETL processes*. In Proc. of the 8th ACM international workshop on Data warehousing and OLAP, (pp. 67–76), ACM, 2005.