On using Weaving Models to Specify Schema Mappings

Sinisa Neskovic, Milica Vuckovic and Nenad Anicic

University of Belgrade, Faculty of Organizational Sciences
str. Jove Ilica 154, 11000 Belgrade, Serbia

Abstract. Weaving models, supported by the ATLAS Model Weaver toolkit within the Eclipse Modeling Framework, has been used for various application scenarios related to model mappings. This paper considers the application of weaving models to specification of data schema mappings. Firstly, a general conceptual framework in the form of an abstract megamodel is introduced. It is used as a reference model which identifies various kinds of models occurring in the context of data schema mappings and precisely defines their roles and mutual mappings. Then, based on the defined conceptual framework, an analysis of the application of weaving models in the context of schema mappings is given. This analysis reveals several issues in the existing approach. The main issue is that weaving models are not enough constrained by their corresponding weaving metamodels and, hence, invalid or semantically meaningless links among schema concepts are allowed. Finally, the paper proposes a solution that overcomes the issues and discusses its advantages and shortcomings.

1 Introduction

Specification of mappings among heterogeneous data schemas has been studied in many different research areas, such as distributed databases [1], data warehouses [3], ontologies [2], model driven development [4,6,7], etc. According to specific needs and characteristics of a particular problem domain, researchers have proposed different approaches and techniques that can be used to specify schema mappings. Without diving into details of each particular approach, it can be generally concluded that most of them rely on a mapping specification formalism, embodied in the form of either a special language or metamodel, which enables expressing and capturing the semantics of correspondences among schema concepts.

In accordance to the motto that “models are everywhere”, the corresponding mapping specification formalism in the context of the model driven engineering (MDE) utilizes (meta) models. One particular solution which has been proposed is based on so called weaving models [5,6,7]. A weaving model is a separate model on its own consisting of elements which represent individual links (i.e. correspondences) among elements of other models (called woven models). A weaving model conforms to a weaving metamodel, which provides the semantics of links specified in a weaving model. A weaving metamodel defines types of links that can occur among elements of woven models, i.e. links specified in a weaving model can be instances

Neskovic S., Vuckovic M. and Anicic N.
On Using Weaving Models to Specify Schema Mappings.
DOI: 10.5220/0003029600460055
In *Proceedings of the 2nd International Workshop on Future Trends of Model-Driven Development (ICEIS 2010)*, page
ISBN: 978-989-8425-10-2

only of types defined in the weaving metamodel. A special core weaving metamodel with generic link types suitable for a range of different application scenarios is also proposed. For each application scenario, the core weaving metamodel is extended with specialized link types that are more suitable for the particular application.

Supported by the ATLAS Model Weaver (AMW) toolkit [12] within the Eclipse Modeling Framework (EMF) environment, the proposed approach has gained a lot of attention in the MDE community lately. It has been reported that weaving models are successfully applied to several MDE related problems, including schema and data mappings problems [5].

However, our experience in the application of weaving models to the problem of schema mappings reveals that there exist some open issues. Namely, the definition of a weaving metamodel is based only on concepts of metameta model (i.e. the ECore metameta model in EMF). It does not rely on concepts of corresponding metamodels of models intended to be woven and, hence, is completely unaware of any semantic rules regarding mappings among concepts of the metamodels in question. As a consequence, link types defined in a weaving metamodel cannot prevent links between elements of woven models which are semantically meaningless, wrong or disallowed. For example, when mapping concepts between an Entity-Relationship (ER) data schema *S1* and a Relational schema *S2*, it is possible to link an entity from the *S1* schema with a column from the *S2* schema. In other words, the weaving model lacks the semantics of mapping rules between ER and Relational schemas, i.e. that entities can be mapped to relational tables only.

This paper proposes a possible solution which overcomes the identified open issues by providing explicit support for mapping rules. The solution is based on a weaving model which serves for definition of mapping rules between schema metamodels. This weaving model is then transformed to a weaving metamodel Link types with OCL constraints. The role of OCL constraints is to restrict links in weaving models to establish only those relationships between schema concepts which are meaningful.

The paper is organized as follows. The next section briefly presents the related work. Section 3 introduces a general conceptual framework in the form of an abstract megamodel, which is used to identify types of models that occur in the context of data schema mappings and to define their roles and mutual mappings. Section 4 analyses the existing practice in utilizing weaving model and discuss shortcomings and open issues. The proposed solution to detected open issues is explained and discussed in Section 5. Section 6 concludes the paper.

2 Related Work

Schema mappings are high-level specifications which express correspondences between two data schemas describing how data sources are organized (structured). The problem of schema mappings is a part of larger problem related to information integration. For example, schema mappings are required for data exchange, i.e. translating data from one data source to other. They are also used for virtual information unification where users are enabled to pose queries over distributed heterogeneous data sources in a uniform and transparent manner.

Many different formalisms and languages for schema mappings are used in various research areas. The basic formalism in relational database integration systems that have been proposed is based on so called source-to-target tuple-generated-dependences (s-t tgds)[16,17]. Special forms of s-t tgds known as local-as-view (LAV) and global-as-view (GAV) specification languages are used for specification of schema mappings when several local schemas are integrated using a global schema.

In the case of information integration when heterogeneous schema languages are being used, several approaches have been proposed [13,14,15]. These approaches are mostly based on a generic metamodel that abstracts concrete schema metamodels. Schema mappings in this case are specified in a language which depends on the used generic metamodel. For instance, in [13] a universal metamodel based on the supermodel is used as a generic metamodel and DATALOG is used for representing schema mappings. In [15], the GeRoMe model and corresponding specification language are used.

In the context of MDE, specifications of data schema mappings can be viewed as a special case of model mappings. Another special case of model mappings are model transformations, which represent a crucial notion in MDE. The MDE community has proposed several model transformation specification languages. For instance, OMG has proposed the QVT language [10], ATL is used in the EMF environment [11], etc. Although transformation specification languages could be used for data schema mappings, they are not designed for such task. Transformation specification languages are used to specify mappings between metamodels (M2 level models) and, consequently, are inconvenient for specification of schema mappings, which are M1 level models.

Model weaving is an approach used for the specification of links between model elements. It is supported by the AMW, a component of the larger Atlas Model Management Architecture toolkit [12]. This approach is conceived with a goal to facilitate a range of different application scenarios, such as tool interoperability, model composition operations, traceability, model alignment, etc [12].

One group of supported application scenarios is related to data mappings. The work from [5] presents the application of weaving model to discovery of model mappings and production of executable operational mappings (including model transformations) which translate from source models to targets. These results are extended by the work in [7], which utilize successive schema matching transformations to generate and refine a sequence of weaving models until a final one is generated, out of which data transformations are produced.

However, despite these successful applications, there exits some open issues which are discussed later in this paper.

3 Conceptual Data Integration Framework

In order to analyze suitability of the weaving modeling approach for the specification of schema mappings, we introduce a conceptual data integration framework. It is presented in Figure 1 in the form of an abstract generic megamodel. A megamodel is a model whose elements are other models [18, 19]. The main purpose of the introduced framework is to identify kinds of models that occur in the context of data schema mappings and to precisely define their roles and mutual mappings. Hence it is expressed in the form of a megamodel.

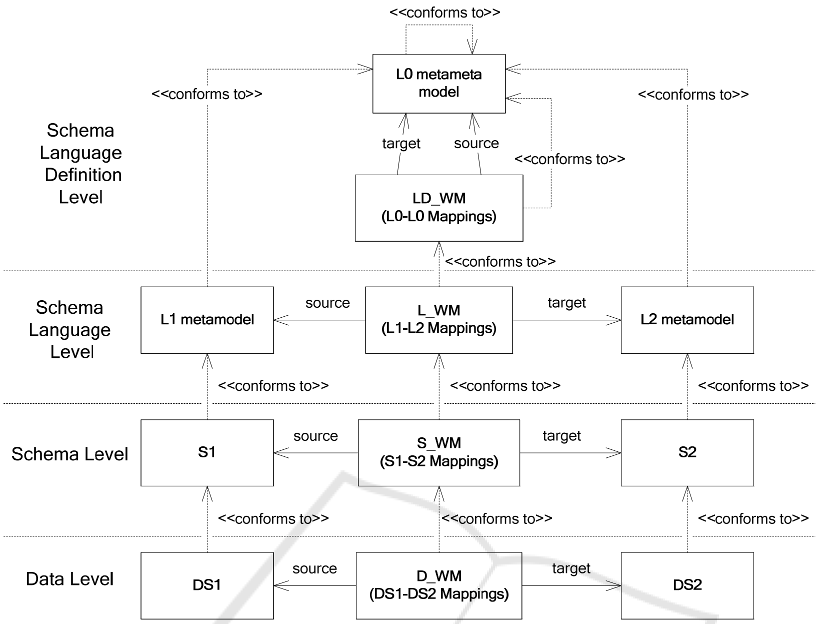

Fig. 1. Conceptual data integration framework.

The framework has 4 abstract levels, which correspond to the levels of the OMG MDA standard [4], but here are named in the manner that is more appropriate for data integration purposes. As it is typical for metamodeling architectures, each level accommodates models which serve as metamodels for other models from the lower abstract level, whilst they must conform to their metamodels from the upper level. The exception is the model at the most abstract level which conforms to itself.

The framework identifies two types of models: (1) ordinary models which are used to describe domain concepts, and (2) weaving models which links elements from other ordinary models[1]. Depending on the abstract level where they reside, the following four kinds of weaving models can be identified:

- *Data Mapping Weaving Models* (D_WM) which specify links at the Data Level, i.e. between data instances stored in different possibly heterogeneous data sources.
- *Schema Mapping Weaving Models* (S_WM) which specify links at the Schema Level, i.e. between schema concepts that are possibly expressed in different schema languages.
- *Language Mapping Weaving Models* (L_WM) which specify links at the Schema Language Level, i.e. specify mappings between concepts of different schema languages.

[1] Mappings between different weaving models are also possible, but their consideration is beyond the scope of this paper.

- *Language Definition Weaving Models* (LD_WM) which specify links at the Schema Language Definition Level, i.e. between concepts of a metameta model used to describe schema languages.

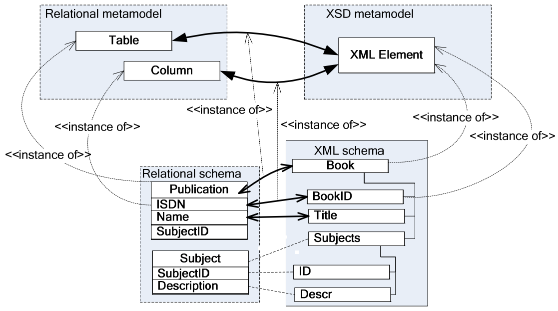

Fig. 2. Links at two different abstract levels.

It is important to note that weaving models must conform to their corresponding metamodels, which are also weaving models. This means that links specified in one weaving model must be instances of links specified in its weaving metamodel. In other words, links specified in the upper abstract level constrain links in the lower level to relate only certain types of model elements. Thus, links serve as mapping rules for the lower level enabling only meaningful links and preventing invalid ones. The example shown in Figure 2 illustrates this for the case of mappings between a relational schema and an XML schema. As it is depicted in the figure, table *Publication* is mapped to XML element *Book* by a link which is an instance of the rule mapping relational tables to XML elements, whereas column ISDN is mapped to XML element *BookID* by a link which is an instance of the rule mapping columns to XML elements.

A special case is LD_WM, which is used to define rules for mappings between schema languages. Since it represents the most abstract weaving model in the framework, it is defined in terms of concepts of a corresponding metameta model *L0*.

4 Open Issues

Model mappings based on weaving models represent an approach in MDE which is supported by the AMW toolkit [9, 11]. It is aimed to support a range of application scenarios where model mappings are involved. Here, we will discuss the approach from the schema mappings perspective only.

AMW supports an extensional mechanism based on the core weaving metamodel that encompasses a set of features (i.e. generic concepts) common in majority of application scenarios. Using extensions one defines a new weaving metamodel based on the core weaving metamodel. A new weaving metamodel typically defines new link types which are specific to a particular application scenario. Defined link types are used in weaving models for relating elements of two woven models.

Using the conceptual data integration framework given in Section 3 as a reference model, the typical application scenario employed by the AMW approach is shown in Figure 3.

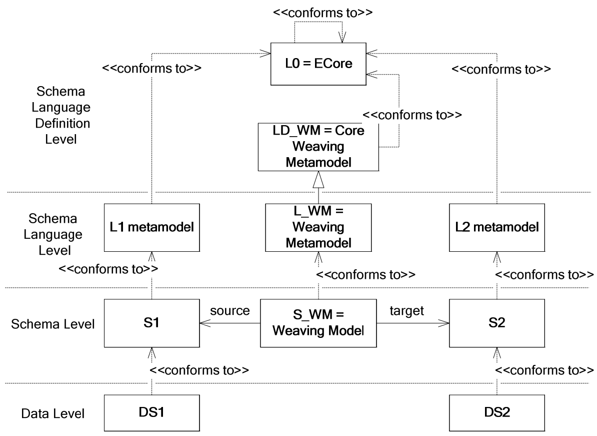

Fig. 3. The AMW Approach.

Here, the ECore metamodel of EMF plays its usual role of the most abstract models *L0*. The role of the LD_WM model in the conceptual framework is played by the core weaving metamodel, which is defined in terms of Ecore concepts. The role of L_WM, used to define mapping rules between different schema languages, is played by a weaving metamodel. It is defined as an extension of the core weaving metamodel. Note that this is different in comparison to our conceptual framework where L_WM is defined as an instance of metamodel. In addition, weaving metamodel in AMW does not specify mappings between concepts of language, but simply defines a new link types. In other words, the semantics is provided only by giving a new name, without specifying schema concept types allowed to be related by this link type.

The role of S_WM, used to specify schema mappings, is played in AMW by weaving model. It is defined as an instance of its corresponding weaving metamodel, which is in accord with the conceptual framework. However, links specified by weaving metamodel can relate any elements from woven models. It is up to a modeler to take care whether such links are meaningful.

Figure 4 illustrates the situation that can happen when a modeler is careless or unaware of semantic mapping rules. The rule *Entity2Table* is specified in the weaving metamodel meaning that entities from ER models are translated to tables in relational schemas. However, as it is shown in the figure, it is possible to create a link which is an instance of the defined rules, but maps an entity to a column.

The Data level from the conceptual framework is not actually supported by AMW. However, application scenarios involving data instances still can be supported by "artificial" lifting of models from the Data and Schema levels for one abstract level up, i.e. by expressing and treating data schemas as metamodels and data instances as

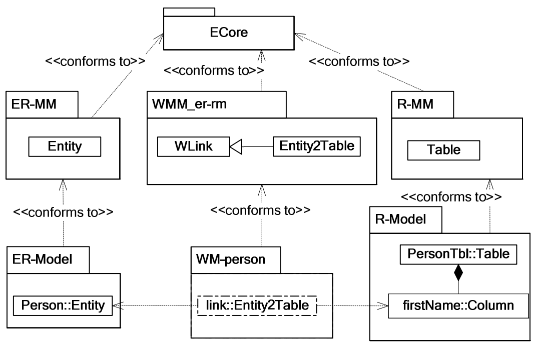

Fig. 4. A semantically invalid link in a weaving model.

M1 models. Such technique is employed in [5]. However, it leads to weaving models that must make up for a lack of models from the Schema Language level, which is lost due to the level lifting. This technique introduces the accidental complexity to the definition of weaving models. Due to limited space, the further detailed discussion of this technique is beyond the scope of this paper.

To summarize the discussion, there exist the following shortcomings and open issues:

- Weaving models may contain invalid specification of schema mappings.
- Weaving models are not adequately constrained by their corresponding weaving metamodels.
- Link ends, which are part of link type definitions, cannot be typed, i.e. specify which concept types they may relate.
- Modelers are supposed to know the semantics of mapping rules and must take care about links they create.

5 Solution

The open issues discussed above can be resolved by better alignment of the AMW approach with the conceptual data integration framework defined in Section 3. Better alignment in the context of schema mappings primarily means that link types defined in weaving metamodels have to constrain links in weaving models to relate those schema concepts that are meaningful. In other words, it is needed to support specifications of mapping rules in weaving metamodels and typed end points of links in weaving models.

This paper proposes a solution which achieves this by using both a weaving model and a weaving metamodel at the Schema Language level to express mapping rules between concepts of data schemas. The proposed solution is given in Figure 5[2].

---

[2] Note that Ecore metameta model and Data level models are deliberately omitted in Figure 5. in order to make the diagram more readable.

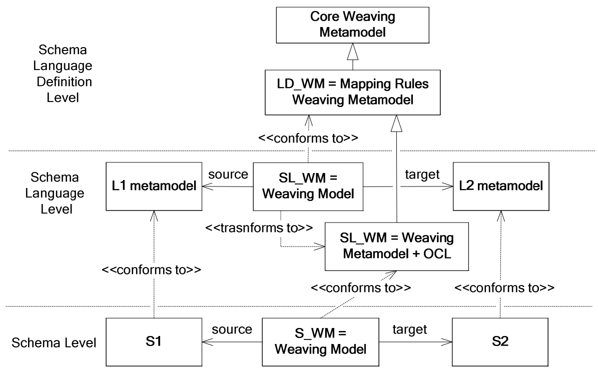

Fig. 5. The Proposed Solution.

A weaving model is used to specify mapping rules by establishing semantically meaningful links between concepts from two schema languages. Due to limitations of the AMW tool, this weaving model cannot be used as a metamodel for weaving models from the Schema level. Hence, this weaving model is automatically transformed into a weaving metamodel enriched with OCL constraints. OCL constraints are integral parts of link type definitions and they specify types of data schemas concepts that can be related by a particular link type. In this way, end points of links in weaving models are allowed to reference only instances of the specified types.

Please note that both the weaving model and generated weaving metamodel contain the same information, but expressed in different representation formats (i.e. structural constrains are expressed as value based ones using OCL). Therefore, both models have the same L_WM role defined in the conceptual framework in Section 3. In addition, both models are related to the same weaving metamodel. Unlike the AMW approach where they are related to the core weaving metamodel, here a new special weaving model playing the LD_WM role is introduced. Also, they are related in a different way. The weaving model conforms to the LD_WM model, whilst the weaving metamodel is defined as its extension.

Figure 6 illustrates the proposed solution for the case of mapping an ER model to a relational schema. The corresponding OCL constraint restricting source and target of the Entity2Table rule is:

```
context Entity2Table inv:
    self.source.oclIsKindOf(Entity) and
    self.target.oclIsKindOf(Table)
```

The main shortcoming of the proposed solution is that it requires an extension of the existing AMW tools. Namely, the AMW Weaving Editor does not support OCL constraints. Therefore, the AMW tool does not restrict links between concepts to relate allowed concepts. In addition, it is unaware of weaving models and metamodels and their role in the specification of mapping rules. Hence, model transformations between them are not supported.

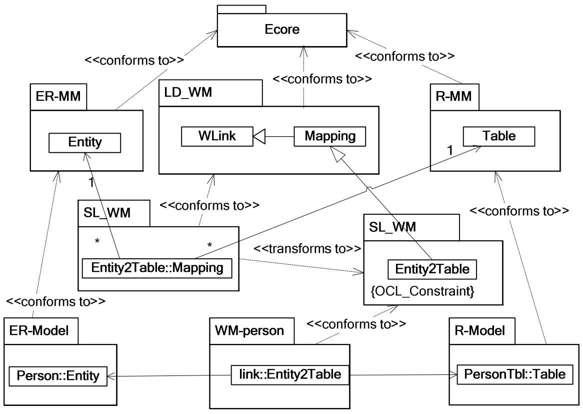

Fig. 6. Illustration of the proposed solution**.**

6 Conclusions

This paper discussed suitability of the weaving model approach in the context of schema mappings. The paper's main contributions are the following:

- A conceptual data integration framework introduced to identify kinds of models that occur in the context of data schema mappings and to precisely define their roles and mutual relationships. It is used as a reference model for analysis of the weaving approach.
- Analysis which reveals that the weaving model approach supported by the AMW tool does not properly support schema mappings, i.e. allows specifications which are semantically meaningless. The main cause is related to the inability of weaving metamodels to properly define semantic mapping rules between schema languages.
- A proposed solution extending the current approach and AMW tool, which is based on introduction of special weaving models and metamodels with OCL constraints. The extensions augment definition of link types in the weaving models and metamodels in order to restrict links between schema concepts.

It can be concluded that the approach based on weaving models has been carefully conceived from both theoretical and technological points of view to be general and flexible enough. This generality and flexibility enables the weaving model approach to be applicable in a wide range of MDE related tasks. However, such generality and flexibility have shortcomings when applied in the context of schema mappings.

Our future work concerns realization of the extensions of the AMW tool proposed here to support schema mappings. We also intend to utilize the introduced conceptual data integration framework to support data level mappings as well. The ultimate goal would be to have a unified and comprehensive methodological approach and tool that would properly support all mapping specification levels of the data integration framework.

References

1. Lenzerini, M.: Data Integration: A Theoretical Perspective. PODS 2002 (2002) 233-246
2. Ehrig, M., Sure, Y.: Ontology Mapping - An Integrated Approach. ESWS (2004) 76-91
3. Cui, Y., Widom, J.: Lineage Tracing for General Data Warehouse Transformations. VLDB J. 12 (2003) 41-58
4. Miller, J., Mukerji, J.: Model Driven Architecture (MDA). OMG Document available at http://www.omg.org; 2001.
5. Didonet Del Fabro, M., Bézivin, J., Jouault, F., Valduriez, P.: Applying Generic Model Management to Data Mapping. In: Benzaken, V. (ed): BDA, Saint Malo, Actes, 2005.
6. Del Fabro. M.D., Valduriez, P.: Semi-automatic Model Integration using Matching Transformations and Weaving Models. SAC (2007) 963-970
7. Del Fabro, M.D., Valduriez, P.: Towards the Efficient Development of Model Transformations Using Model Weaving and Matching Transformations. Software and System Modeling 8(3) (2009) 305-324
8. Eclipse Modeling Framework Project (EMF). http://www.eclipse.org/modeling/emf/
9. Object Management Group (OMG): Object Constraint Language OMG Available Specification Version 2.0, May 2006.
10. OMG, Revised Submission for MOF 2.0 Query/View/Transformations RFP (ad/2002-04-10), OMG Document ad/2005-07-01 (2005)
11. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I.: ATL: A model transformation tool. Sci. Comput. Program. (SCP) 72(1-2) (2008) 31-39
12. Del Fabro, M.D., Bézivin, J., Valduriez, P.: Weaving Models with the Eclipse AMW plugin. In: Eclipse Modeling Symposium, Eclipse Summit Europe 2006.
13. Atzeni, P., Cappellari, P., Torlone, R., Bernstein, P.A., Gianforme, G: Model-independent schema translation. The VLDB Journal 17 (2008) 1347–1370
14. Atzeni, P., Gianforme, G., Cappellari, P.: A Universal Metamodel and Its Dictionary. A. Hameurlain et al. (Eds.): Trans. on Large-Scale Data- & Knowl.-Cent. Syst. I, LNCS 5740, (2009) 38-62
15. Kensche, D., Quix, C., Li, X., Li, Y., Jarke, M.: Generic Schema Mappings for Composition and Query Answering. Data & Knowledge Engineering 68 (2007) 599-621
16. ten Cate, B., Kolaitis, P.G.: Structural Characterizations of Schema-mapping Languages. Commun. CACM (CACM) 53(1) (2010) 101-110
17. Kolaitis, P.G.: Schema Mappings, Data Exchange, and Metadata Management. Symposium on Principles of Database Systems (PODS) (2005) 61-75
18. Bezivin, J., Jouault, F., Valduriez, P.: On the Need for Megamodels. In: Proceedings of the OOPSLA/GRE: Best Practices for Model-Driven Software Development workshop, 19th Annual ACM OOPSLA 2004.
19. Favre, J-M., Nguyen, T.: Towards a megamodel to model software evolution through transformations. SETRA Workshop, Elsevier ENCTS, 2004, pp. 59-74.