A Hybrid Model Driven Development Framework for the Multidimensional Modeling of Data Warehouses*

Jose-Norberto Mazón
Lucentia Research Group
Dept. of Software and Computing Systems
University of Alicante, Spain
jnmazon@dlsi.ua.es

Juan Trujillo
Lucentia Research Group
Dept. of Software and Computing Systems
University of Alicante, Spain
jtrujillo@dlsi.ua.es

ABSTRACT

Developing a multidimensional (MD) model of a data warehouse (DW) is a highly complex, prone to fail, and time consuming task, due to the fact that (i) the information needs of decision makers and the available operational data sources that will populate the DW must both be considered in a conceptual MD model, and (ii) complex mappings must be performed to obtain an implementation of this conceptual MD model. However, no significant effort has been made to take these issues into account in a systematic, well structured and comprehensive development process. To overcome the lack of such a process, a framework based on the Model Driven Architecture (MDA) is proposed for the development of a hybrid MD model at the conceptual level and for the automatic derivation of its logical representation. Also, a running example is shown throughout this paper.

1. INTRODUCTION

Data warehouse (DW) systems provide a multidimensional (MD) view of huge amounts of historical data from operational sources, thus supplying useful information for decision makers to improve a business process in an organization. The MD paradigm structures information into facts and dimensions. A fact contains the interesting measures (fact attributes) of a business process (sales, deliveries, etc.), whereas a dimension represents the context for analyzing a fact (product, customer, time, etc.) by means of hierarchically organized dimension attributes. MD modeling requires specialized design techniques that resemble the traditional database design methods [16]. First, a conceptual design phase is performed whose output is an implementation-independent and expressive MD model for the DW. A logical design phase then aims to obtain a technology-dependent model from the previously defined conceptual MD model. This logical model is the basis for the implementation of the DW. Therefore, there are two cornerstones in MD modeling: the development of a conceptual MD model and the derivation of its corresponding logical representation.

On one hand, with regard to conceptual design, current approaches usually embrace one of the following perspectives: (i) *data-driven*, in which the conceptual MD model is based on a detailed analysis of the data sources; while information requirements are only considered later when the implemented MD model is queried, and (ii) *requirement-driven*, in which the conceptual MD model design is based on the information needs of decision makers, thus considering the available data sources later when the DW is populated. Nevertheless, although data-driven approaches simplify the process of properly populating the DW, user needs and expectations may not be satisfied by the MD model designed because these approaches do not provide enough mechanisms through which to analyze and understand the decision making process supported by the DW. Moreover, data-driven approaches do not provide mechanisms to highlight the important parts of the data sources, thus wasting resources by specifying unneeded information structures in the MD model. Conversely, requirement-driven approaches involve decision makers in the development of the MD model, but lack mechanisms through which to formally match the data sources with information requirements in early stages of the development, thus making it highly complex to populate the DW in a proper manner, as the correspondence between the elements of the MD model and their counterparts in the data sources may not be obvious. In order to overcome these drawbacks, we argue that the most promising solution consists of formally considering both the data sources and the requirements in a hybrid approach.

On the other hand, once a conceptual MD model is designed, another important issue must be faced: obtaining a logical representation of the conceptual MD model. This is a tedious and repetitive task which requires a high degree of automatization if failures in the implementation of the DW are to be avoided.

Bearing these considerations in mind, two tasks must be considered in the development of the DW if the previously-stated problems are to be overcome: (i) the formal reconciliation of the operational data sources and the information needs of decision makers at the conceptual level, and (ii) the formal specification of transformations which will allow designers to obtain the most suitable logical representation of

*Work supported by the ESPIA (TIN2007-67078) project from the Spanish Ministry of Education and Science, and and by the QUASIMODO project (PAC08-0157-0668) from the Castilla-La Mancha Ministry of Education and Science. Jose-Norberto Mazón is funded by the Spanish Ministry of Education and Science under a FPU grant (AP2005-1360).

the developed conceptual MD model in an automatic manner. Several previous works have separately covered these tasks [7, 8, 9, 10, 11, 12]. In this paper, the main contribution is to propose a Model Driven Development (MDD) process to elegantly combine these previous works into an integrated framework. MDD is useful to this aim because it offers mechanisms to both manage the integration of models and to define the transformations between them. Other works have, in fact, taken advantage of using MDD in database development (such as [20]). However, to the best of our knowledge, our work is the first to tackle MD modeling problems and overcome them by using MDD.

The remainder of this paper is structured as follows. Sect. 2 presents current approaches towards the MD modeling of DWs, stressing how our work contributes to the state-of-the-art. Sect. 3 describes our overall MDD approach towards MD modeling. A running example is used to clarify theoretical concepts. Our conclusions are presented in Sect. 4.

2. RELATED WORK

Research in DW modeling has been tackled from several points of view. For example, the design of ETL processes [19] or the development of formal models for data analysis [18]. The focus of this paper is on research that addresses MD modeling of the DW repository.

Current approaches towards conceptual MD modeling focus on providing mechanisms with which to specify an implementation-independent MD model. These approaches can be classified into data-driven [4] or requirement-driven approaches [15]. Data-driven approaches are based on analyzing the operational data sources, while requirements are considered later when the data is about to be analyzed. Requirement-driven approaches focus on the information needs of decision makers, and data sources are only taken into account later when the data is loaded into the DW.

Surprisingly, few approaches advocate the consideration of both data sources and information requirements in the early stages of development [3]. Importantly, these approaches rely on a manual analysis of data sources to reconcile them with the requirements, which could be unfeasible in real-life situations, where the operational data sources are huge. Therefore, an automatization of this reconciliation process is faced up to as a key issue in MD modeling [12].

Moreover, approaches for conceptual MD modeling either do not give mechanisms with which to derive the logical representation [1] or they only describe a set of informal guidelines through which to derive a logical representation with a low level of automatization [4]. Other works [7] have pointed out the need of having a set of transformation rules for deriving logical schemas. However, until now, mechanisms with which to define formal transformations to automatically derive every possible logical representation from the conceptual MD model are not provided, and obtaining logical models thus becomes a time-consuming, tedious and prone to fail task for designers.

To overcome these problems, a hybrid approach for MD modeling is described in this paper. This approach (i) combines both data-driven and requirement-driven strategies in an integrated fashion (i.e. reconciling data sources and decision makers' information needs at the conceptual level) in such a way that the DW meets decision makers' needs and simultaneously agrees with data sources, and (ii) provides a repository of formal transformations through which to include the knowledge about how to automatically obtain a suitable logical representation from a conceptual MD model, thus ameliorating the tedious and prone to fail task of designers who can save time and effort in obtaining the implementation of the DW.

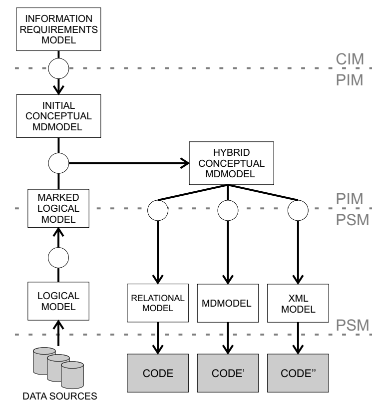


Figure 1: Hybrid MDA-based framework for DWs

3. HYBRID MULTIDIMENSIONAL MODEL DRIVEN DEVELOPMENT

Several parts of our MD modeling approach have been separately defined in our previous work [7, 8, 9, 10, 11, 12]. Now, in this paper, thanks to the integration mechanisms provided by MDD, each piece is fitted into an overall framework (see Fig. 1). The novelty of this framework is that it considers a hybrid viewpoint for MD modeling in a systematic, well structured and comprehensive way, whilst a set of formal transformations are simultaneously established in order to obtain the final implementation of the MD model in an automatic manner. Concretely, our framework is based on the Model Driven Architecture (MDA) [13] proposed by the OMG as a standard with which to carry out the MDD. As is shown in Fig. 1, a conceptual MD model of the DW (Platform Independent Model, PIM) is developed from an information requirements model (Computation Independent Model, CIM) obtained from decision makers [9]. This initial PIM must then be reconciled with the data sources [12, 10], thus obtaining a hybrid PIM. Each element of these data sources must be previously marked with its MD counterparts. Moreover, several logical models can be derived from this hybrid PIM as Platform Specific Models (PSMs), by taking into account different deployment platforms (relational, multidimensional, etc.). The relations between models are implemented by using the Query/View/Transformations (QVT) language [13]. These relations are shown in Fig. 1 as circles. Finally, several transformations have been defined in order to obtain the code for the implementation of the MD model according to each PSM. These transformations have been established by means of the Model to Text Transformation (Mof2Text) language [13].

A plugin that supports our approach has been developed with the *Eclipse Modeling Framework*[1]. Our running example has been implemented by using this plugin.

[1]`http://www.eclipse.org/modeling/emf/`

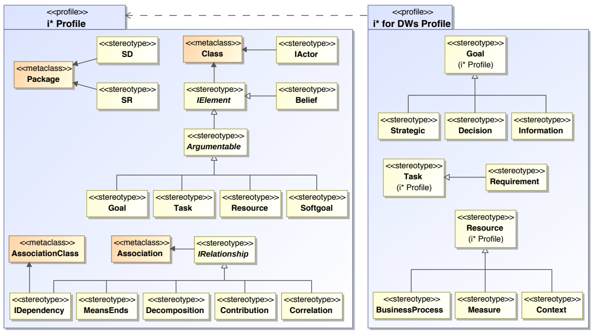


Figure 2: UML profiles for i* modeling in DWs

3.1 Multidimensional CIM

An explicit requirement analysis stage is needed in order to model the information requirements of decision makers and to derive a suitable conceptual MD model which meets the real needs of decision makers, thus increasing the success of a DW project. Decision makers who use DWs often ignore how to suitably describe information requirements, since they are concerned rather with the goals which the DW helps to fulfil. Therefore, a requirement analysis phase for DWs must ideally start by discovering decision makers' goals. The information requirements can be discovered more easily from these goals.

Goals related to the DW can be specified on three levels [9]: *Strategic goals*, which are main objectives of the business process: e.g. *"increase sales"*. *Decision goals* aim at taking the appropriate actions to fulfil a strategic goal, e.g. *"open new stores"*. Finally, *information goals* are related to the information required by a decision goal to be achieved; e.g. *"analyze customer purchases"*. Once these goals are defined, information requirements can be directly obtained from the information goals. The various MD elements, such as *facts* or *dimensions*, will be discovered from these information requirements in order to specify the corresponding conceptual MD model of the DW.

In order to model goals and information requirements in a CIM, the UML (Unified Modeling Language) [13] has been used together with the *i** modeling framework [21]. Specifically, we have developed [9] (i) a UML profile for *i**, in order to integrate it within our MDA framework; and (ii) a UML profile which adapts *i** to the DW domain. Both profiles are described in Fig. 2.

The *i** modeling framework provides mechanisms with which to represent the various DW actors, their dependencies, and with which to structure the business goals that the organization wishes to achieve. Decision makers' goals are defined by using the *Strategic*, *Decision*, and *Information* stereotypes. Information requirements (*Requirement*) are derived from information goals and are represented as stereotyped tasks. Furthermore, the requirement analysis for DWs necessitates the addition of certain MD concepts (in the sense of [3]). Therefore, the following concepts are added to the CIM as stereotyped resources: business processes related to the goals of decision makers (*BusinessProcess* stereotype), relevant measures related to decision makers' information requirements (*Measure*), and the contexts needed to analyze these measures (*Context*). Additionally, foreseen relations between the contexts of analysis are modeled. For instance, the city and the country contexts are related because cities can be aggregated in countries. In order to model these relationships, we have used the (shared) aggregation relationship of UML.

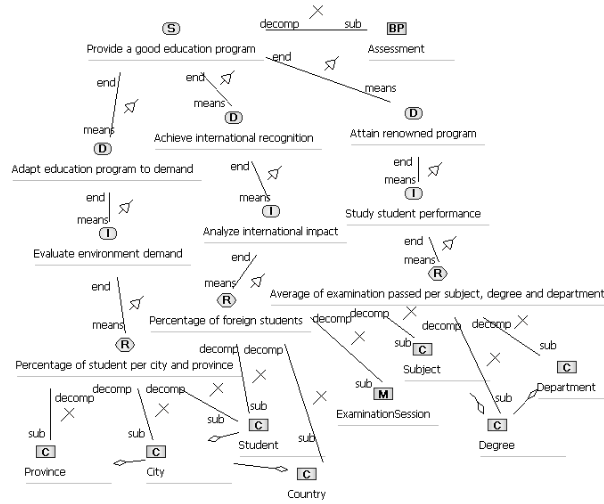


Figure 3: CIM for our running example

An example of how to use our approach is illustrated through a running example based on the strategic educational plan of the University of Alicante[2]. This plan determines goals and actions which must be undertaken in order to provide a high-quality education program. This running example specifically focuses on developing a DW that will support decision making in the *"assessment"* process. This process is related to one main actor, the *"education manager"*, via the strategic goal *"provide a good education program"*. Three different decision goals are derived from this strategic goal: *"adapt education program to demand"*, *"achieve international recognition"*, and *"attain renowned program"*. The following information goals have been obtained from each of these decision goals : *"evaluate environment demand"*, *"analyze international impact"*, and *"study student performance"*. The derived information requirements are as follows: *"percentage of students per city and province"*, *"percentage of foreign students"*, *"average of examination passed per subject, degree and department"*. Furthermore, the necessary measures and contexts of analysis are associated with the information requirements. The sole measure is *"examination session"*, and the elements that represent the context of analysis are *"student"*, *"city"*, *"province"*, *"country"*, *"subject"*, *"degree"*, and *"department"*. Several of these contexts of analysis are related to each other in order to aggregate data conveniently. Each of these elements is defined in a CIM according to the UML profile for *i** (see Fig. 3).

In summary, if a CIM is to be properly defined, then several steps must be followed: (i) discovering the actors (i.e. the decision makers), (ii) discovering their goals according to

[2] `http://www.ua.es`

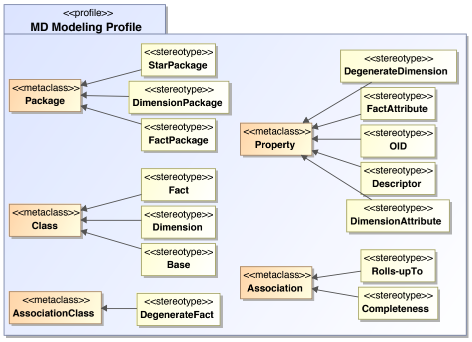


Figure 4: UML profile for MD modeling of DWs

the classification previously described, (iii) deriving information requirements from information goals, and (iv) obtaining the MD concepts related to the information requirements.

3.2 Initial Multidimensional PIM

Once goals and information requirements are specified in a CIM, a conceptual MD model that supports them must be derived in a PIM independently of any database technology. Therefore, within our MDA approach several QVT transformation rules are applied to obtain this initial PIM, thus assuring traceability between goals and information requirements in the CIM and MD elements in the PIM.

The definition of the PIM is based on our UML profile for conceptual MD modeling presented in [7]. This profile contains the stereotypes and constraints necessary to elegantly represent main MD properties at the conceptual level by means of a UML class diagram (see Fig. 4).

To automatically obtain a PIM from the CIM we have developed a set of QVT relations including four main rules [9]: (i) a hierarchy of goals together with a business process is transformed into a fact, (ii) measures, which are linked to an information requirement, are transformed into fact attributes within the corresponding fact, (iii) contexts for analyzing a fact are transformed into dimensions, and (iv) initial hierarchies can be discovered from the relations between these contexts of analysis.

By applying this QVT transformation to the CIM of our running example, the initial PIM (shown in Fig. 5) is obtained: a *Fact* class *Assessment* is created with a *FactAttribute* property *ExaminationSession*; two *Dimension* classes, and their hierarchy levels (*Base* classes), are also created according to the contexts of analysis defined in the CIM.

3.3 Hybrid Multidimensional PIM

As has previously been described, the initial PIM is directly derived from the CIM, thus ensuring that the DW will be useful in fulfilling decision makers' goals. However, this initial PIM is defined without taking the operational data sources into account, and it may not agree with these sources because decision makers may have a limited view of them. Due to this fact, the initial PIM might not be faithful (it may not be properly populated from data sources) nor complete (it may not capture the analysis potential provided by the data sources). Therefore, if these flaws are to be avoided, then this initial PIM must be checked against

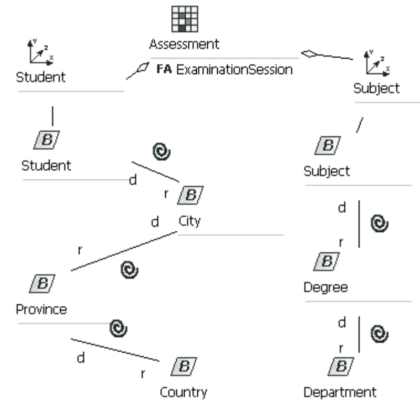


Figure 5: Initial PIM for our running example

the available data sources.

Interestingly, several MD normal forms have been developed [6] to reason, in a rigorous manner, about several desirable properties of a conceptual MD model derived from operational data sources (among others, faithfulness and completeness). Hence, we have developed a set of QVT relations based on these MD normal forms [12] to ensure that an initial PIM is faithful and complete with respect to the source databases, thus obtaining a hybrid PIM.

The approach through which we obtain a hybrid PIM consists of two main phases. The first is based on considering the design of the MD model as a software modernization task [10]. The aim of this task is to link MD concepts to elements of the data sources, thus facilitating the subsequent phase [17]. This first phase starts by using data reverse engineering mechanisms in order to obtain a logical representation of data sources. The motivation is that operational data sources are truly legacy systems, therefore the documentation is not generally available, it cannot be obtained, or it is too complex to be easily understandable through a manual analysis [2]. This first phase concludes with the application of a set of QVT rules to identify MD concepts (fact, dimension, measure and so on) in the logical representation of data sources in order to obtain a marked logical model.

The second phase consists of reconciling the initial PIM with the marked logical model of data sources by using a set of QVT relations based on MD normal forms [12], thus obtaining a hybrid PIM. These relations focus on detecting the functional dependencies (FDs) implied by the initial PIM and by the source databases. Specifically, *faithfulness* is checked by ensuring that the FDs implied by the initial PIM are a subset of those observed in the source databases (otherwise, some source data cannot be represented under the MD model), while *completeness* is enforced by ensuring that FDs among dimension levels contained in the source databases are represented as *Rolls-upTo* associations in the PIM and that FDs among sets of measures contained in the source databases are represented via derivation formulas in the PIM (otherwise analysis potential is lost in the MD model). Furthermore, MD normal forms ensure that each measure is assigned to a fact at the "right" level of detail (without redundancies). The set of developed QVT relations

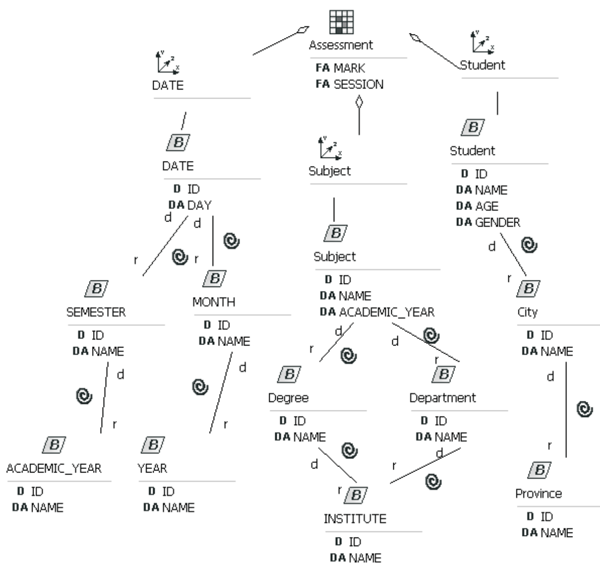


Figure 6: Hybrid PIM for our running example

based on the MD normal forms enforces these properties by removing, adding or modifying elements in the initial PIM, thus obtaining the hybrid PIM.

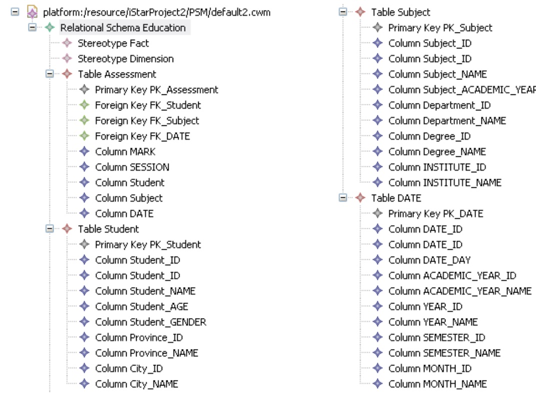
For example, to ensure faithfulness, *ExaminationSession*, *Country* and the *Rolls-upTo* association between *Degree* and *Department* are removed from the PIM of our running example (see Fig. 6) because they have no corresponding counterparts in the data source model. On the other hand, to ensure completeness, the hybrid PIM of Fig. 6 shows new useful elements which are part of the data sources but which do not appear in the initial PIM (e.g. the *Dimension* class *Date* or the *Base* class *Institute* which is the right way to relate *Degree* and *Department* to each other).

The great benefit of this hybrid PIM lies in that it faithfully represents the data sources and completely captures their analysis potential whilst decision makers' information requirements are simultaneously fulfilled.

3.4 Multidimensional PSM

A PSM represents the model of the same system specified by the PIM, but it also captures how that system makes use of one specific platform or technology. In MD modeling, platform-specific means that the PSM is specially designed for a kind of database technology: relational technology (relational database to store multidimensional data), multidimensional technology (structures the data directly in multidimensional structures) or XML technology (semistructured database to store multidimensional data).

In our approach, each PSM is modeled by using CWM (Common Warehouse Metamodel) [13]. Basically, CWM is a metamodel definition for interchanging DW specifications between different platforms and tools. CWM provides a set of metamodels that are comprehensive enough to be able to model an entire DW including data sources, ETL processes, MD modeling, relational implementation of a DW, and so on. These metamodels are intended to be generic, external representations of metadata in order to ensure their interchange among different platforms and tools. Specifically, the Resource layer of CWM contain several metamodels with which to represent in a PSM the structure of data according to different technologies: (i) *relational metamodel* to represent every aspect of relational databases (tables, columns, primary keys, etc.); (ii) *multidimensional metamodel* to represent commonly used multidimensional data structures; or (iii) *XML metamodel* to represent common metadata describing XML data resources.



(a) (b)

Figure 7: PSM for our running example

Following our running example, the PSM corresponds to the most common logical representation of MD models: the relational *star schema* [5]. This schema consists of a central fact table with a composite key which is joined to several dimension tables, each with a single primary key (see Fig. 7).

A set of QVT relations has been defined to establish formal and automatic transformations between the conceptual MD model and the star schema, i.e. between our hybrid PIM and the PSM. For the sake of understanding, these relations are informally described (the corresponding formal QVT relations are shown in [11]):

- Each dimension in the PIM is transformed into a table in the PSM. The descriptor of the dimension is the primary key of the new table.
- Once the dimensions have been transformed, each fact is transformed into a table. This new table has several columns which are foreign keys to every dimension table previously created. Moreover, the primary key of this fact table is a compound of every column that is a foreign key to a dimension table.
- Each dimension attribute belonging to a hierarchy level is transformed into a column in the PSM. These columns belong to the dimension table that is related to the hierarchy level.
- The fact attributes that belong to the fact class are translated into columns of the respective fact table.
- Degenerate dimensions belonging to a fact class are transformed into columns of the fact table. Moreover, degenerate dimensions take part of the primary key of the fact table.

- Degenerate facts are presented when there is a many-to-many relationship between fact and dimension. They are represented as bridge tables in the PSM.

It is worth noting that, although our PSM is assumed to be a star schema, other transformations can be specified in order to obtain different kinds of PSMs, e.g. a PSM directly based on multidimensional technology (as shown in [8]).

Finally, a set of Mof2Text transformations have been developed to obtain the corresponding code for each PSM. For instance, a relational PSM would derive SQL code. However, since the CWM metamodels are close to their respective technologies, deriving code is a straightforward task which is not dealt with in this paper.

4. CONCLUSIONS AND FUTURE WORK

A DW is an integrated collection of historical data which supports management's decisions. According to this definition, in the development of a MD model for the DW, it is not only important to take into account the information needs of decision makers (requirement-driven approaches), but also the existing data sources that will populate the DW (data-driven approaches). Therefore, formal mechanisms are needed to integrate these two points of view in a hybrid approach. Furthermore, the MD modeling of the DW resembles the traditional database design methods [16] because it must be structured into a variety of steps during which a conceptual design phase is performed, whose results are transformed into a logical data model as the basis for schema implementation. This manner of proceeding claims for the automation of these transformations.

In this paper, a framework for dealing with these issues has been presented in order to obtain a major benefit: the systematic, well structured and comprehensive development of a hybrid MD model at the conceptual level and the automatic derivation of its logical representation. Specifically, our hybrid MD modeling framework aligns with MDA by (i) specifying the information requirements in a CIM, (ii) describing how to derive an initial PIM for the MD modeling from the CIM, (iii) reconciling this PIM with the information provided by the available data sources which will populate the DW, thus obtaining a hybrid PIM, (iv) using CWM to build PSMs tailored to various database technologies, and (v) formally establishing QVT relations between these models and Mof2Text transformations in order to obtain the corresponding code. Therefore, our framework allows designers to decrease the inherent complexity of DW development, thus saving time and effort.

Our immediate future work will focus on considering complex MD structures such as heterogeneous dimension hierarchies [14] in the PIM and how they are translated into a PSM, thus avoiding inaccurate queries.

5. REFERENCES

[1] A. Abelló, J. Samos, and F. Saltor. YAM^2: a multidimensional conceptual model extending UML. *Inf. Syst.*, 31(6):541–567, 2006.

[2] R. Alhajj. Extracting the extended entity-relationship model from a legacy relational database. *Inf. Syst.*, 28(6):597–618, 2003.

[3] P. Giorgini, S. Rizzi, and M. Garzetti. GRAnD: A goal-oriented approach to requirement analysis in data warehouses. *Decis. Support Syst.*, 45(1):4-21, 2008.

[4] M. Golfarelli, D. Maio, and S. Rizzi. The Dimensional Fact Model: A conceptual model for data warehouses. *Int. J. Cooperative Inf. Syst.*, 7(2-3):215–247, 1998.

[5] R. Kimball and M. Ross. *The Data Warehouse Toolkit*. Wiley & Sons, 2002.

[6] J. Lechtenbörger and G. Vossen. Multidimensional normal forms for data warehouse design. *Inf. Syst.*, 28(5):415–434, 2003.

[7] S. Luján-Mora, J. Trujillo, and I.-Y. Song. A UML profile for multidimensional modeling in data warehouses. *Data Knowl. Eng.*, 59(3):725–769, 2006.

[8] J.-N. Mazón, J. Pardillo, and J. Trujillo. Applying transformations to model driven data warehouses. In *DaWaK*, volume 4081 of *Lecture Notes in Computer Science*, pages 13–22. Springer, 2006.

[9] J.-N. Mazón, J. Pardillo, and J. Trujillo. A model-driven goal-oriented requirement engineering approach for data warehouses. In *ER Workshops*, volume 4802 of *Lecture Notes in Computer Science*, pages 255–264. Springer, 2007.

[10] J.-N. Mazón, and J. Trujillo. A Model Driven Modernization Approach for Automatically Deriving Multidimensional Models in Data Warehouses. In *ER*, volume 4801 of *Lecture Notes in Computer Science*, pages 56–71. Springer, 2007.

[11] J.-N. Mazón and J. Trujillo. An MDA approach for the development of data warehouses. *Decis. Support Syst.*, 45(1):41-58, 2008.

[12] J.-N. Mazón, J. Trujillo, and J. Lechtenbörger. Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data Knowl. Eng.*, 63(3):699–725, 2007.

[13] Object Management Group. `http://www.omg.org`.

[14] E. Pourabbas, and M. Rafanelli. Hierarchies and Relative Operators in the OLAP Environment. *SIGMOD Record 29(1)*: 32-37 (2000)

[15] N. Prakash, Y. Singh, and A. Gosain. Informational scenarios for data warehouse requirements elicitation. In *ER*, Volume 3288 of *Lecture Notes in Computer Science*, pages 205–216, Springer 2004.

[16] S. Rizzi, A. Abelló, J. Lechtenbörger, and J. Trujillo. Research in data warehouse modeling and design: dead or alive? In *DOLAP* 2006, pages 3–10.

[17] I.-Y. Song, R. Khare, and B. Dai. SAMSTAR: a semi-automated lexical method for generating star schemas from an entity-relationship diagram. In *DOLAP* 2007, pages 9–16.

[18] N. Spyratos. A functional model for data analysis. In *FQAS* 2006, pages 51-64.

[19] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, S. Skiadopoulos. A generic and customizable framework for the design of ETL scenarios. *Inf. Syst. 30(7)*: 492-525 (2005)

[20] B. Vela, E. Fernández-Medina, E. Marcos, and M. Piattini. Model driven development of secure XML databases. *SIGMOD Record 35(3)*: 22-27 (2006)

[21] E. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *RE'97*, pages 226-235.