ELSEVIER

Exploring Model Driven Architecture Approach to Design Star Schema for a Data Warehouse

Anil Sharma[a], Manu Sood [b]

[*]*Department of Computer Applications, PCTE Group of Institutes, Ludhiana, Punjab, INDIA*
[b†] *Department of Computer Science, Himachal Pradesh University, Shimla, Himachal Pradesh, INDIA*

Abstract

Data Integration Technologies have experienced explosive growth in the last few years, and data warehousing has played a major role in this context. This has become one of the most important applications of database technology today. A large number of data warehousing methodologies and tools are available to support the growing market needs but it is generally agreed that warehouse design is a key problem and that multidimensional data models and star schemata are the two focus areas. The Model Driven Architecture (MDA) is a framework defined by the Object Management Group (OMG) for model driven software development. It is possible to have architecture in MDA that will be language, vendor and middleware neutral. This approach places the emphasis on models, provides high levels of abstractions during development and enables significant decoupling between Platform Independent Models (PIMs) and Platform Specific Models (PSMs). Although a lot of research has been carried out on how to design a data warehouse, there is no consensus on a design method. This paper proposes a Model Driven Architecture approach to design star schema for a data warehouse and highlights the significance of using this approach.

Keywords: Data Warehouse, Model Driven Architecture (MDA), Platform Independent Model (PIM), Platform Specific Model (PSM), Star Schema

1. Introduction

A data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management decisions. Data warehousing populates data from heterogeneous sources into a central data warehouse (DW) by Extract-Transform-Load (ETL) at regular intervals of time, e.g., monthly, weekly, or daily. A data warehouse encompasses all of the technologies used to both store and access the data. These technologies include the database management system (DBMS), query tools, report tools, data transformation tools and network infrastructure necessary to support the technology. Taking data stored in on-line transaction processing (OLTP) systems, aggregating and summarizing the data, and then presenting this information to business users, allows corporations to achieve greater productivity and allows business analysts to take more informed decisions. Because of these benefits, more and more organizations are looking forward to implement data warehouse technologies to achieve a competitive advantage [1]. Our capabilities of generation, collection and storage of data have been increasing rapidly since last few decades. This explosive growth in stored data has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amounts of data into useful information and knowledge. Hence, the data integration technologies have experienced explosive growth, and data warehousing has played a major role in this context. It has become one of the most important applications of database technology today. In fact, for the support of growing market, a whole lot of data warehousing methodologies and tools have been developed. But the design of a quality and efficient data warehouse still remains an issue for the researchers. Although quite an effort made by various researchers can be found in literature on how to design a data warehouse, yet there is no consensus on a common design method. Since star schema can be a critical component in the design of a data warehouse, the authors in this paper propose a Model Driven Architecture approach to design a star schema for Data Warehouse.

MDA is an approach to system specification and interoperability, based on the use of formal models. In MDA, platform-independent models (PIMs) are initially

[*] anilmanu2001@hotmail.com
[†] soodm_67@yahoo.com

expressed in a platform-independent modeling language, such as UML. This PIM is subsequently translated to a platform-specific model (PSM) by mapping the PIM to some implementation language or platform using formal rules. At the core of the MDA concepts are a number of important OMG standards: The Unified Modeling Language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI), and the Common Warehouse Metamodel (CWM). These standards define the core infrastructure of the MDA, and have greatly contributed to the current state-of-the art of systems modeling. MDA is aimed at separating business logic from its application logic. The business logic exhibits lesser complexity compared to that inherent in the underlying implementation technologies of application logic. This is so because the business logic presents the static and dynamic aspects of a system at a higher level of abstraction with the implementation details hidden. This abstraction is called Platform Independent Model (PIM). The MDA also supports Platform Specific Model (PSM), which contains enough implementation information to convert it to particular source codes.

2. Data Warehouse Architecture

Data warehouses (DW) play a central role in most of the current decision support systems since they provide crucial business information to improve strategic decision making processes. However, building a DW is still a challenging and complex task because it concerns many functioning units and can often involve many groups with different skill sets and targets. Moreover, the architecture of a DW is usually depicted as a multi-layer system in which data from one layer is derived from data of the previous layer [2] as presented in figure 1.

This architecture has encouraged different methods and approaches for designing different parts of DWs. Yet, these methods fall short of dealing with the design of a DW in an integrated manner. Instead, most of these methods focus on providing partial layer-wise solutions to certain issues, such as ETL (Extraction-Transformation-Load) processes or multidimensional (MD) modeling etc. Therefore, as a result, many problems derived from interoperability and integration of various layers have cropped up. To overcome these problems, a DW development method based on the UML (Unified Modeling Language) [3] and the UP (Unified Process) have been defined. These methods address the design of the whole DW from operational data sources to the final implementation by using several UML profiles which have been developed to adapt UML to certain aspects of the DW design, such as MD modeling [3], modeling of ETL processes [4], modeling data mappings between data sources and targets [5], and physical modeling of the DW [6].

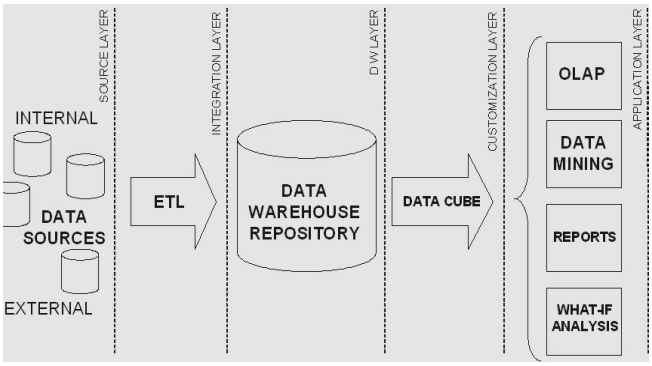

Fig. 1: Data warehouse layers

In this paper, the authors are suggesting to align the whole process of the DW development with MDA for the generation of star schema for the DW. As shown in figure 2, first a CIM has to be developed in order to acquire user requirements. Secondly, various PIMs have to be designed by using several UML profiles mentioned above. These PIMs are conceptual models of each part of the DW itself without considering any aspect of the implementation in a concrete target platform. Normally, PIMs are manually generated from CIM [7]. Then, the resulting PIM can be automatically transformed into several PSMs depending of the required target platform. These transformations are formally specified by using a standard called QVT (Query/View/Transformation) [8]. Finally, the necessary code to implement the DW can be obtained from PSMs.

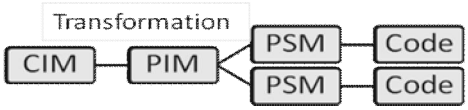

Fig. 2: Model Driven Architecture framework

The most critical issue in MDA is the transformation between a PIM and a PSM. In fact, OMG defines MOF 2.0 Query/View/Transformation (QVT), an approach for expressing model transformations [8]. This is a standard under development for defining transformation rules between MOF-compliant models (e.g. UML models). It defines a hybrid specification for transformations. The authors here wish to point out that QVT is not restricted to PIM-PSM transformations, but PIM-PIM or PSM-PSM transformations can also be developed, provided their metamodels are MOF-compliant.

3. Related Work

Various approaches for the conceptual and logical design of DW systems have been proposed in the last few years. In this section, the authors present a brief overview of some of the most well-known approaches.

Kimball in [1] has presented different case studies of data marts (DM). The use of the star schema and its different variations (snowflake and fact constellation) has been proposed to build a corporate DW by integrating the design of several DMs. Although the authors consider this work as a fundamental reference in the MD field, but a formal method for the design of DWs is missing. Furthermore, the author has a focus on logical model only, and no conceptual model has been proposed to be developed.

Golfarelli at el. in [9] propose the Dimensional-Fact Model (DFM), a particular notation for the DW conceptual design. Moreover, they also propose how to derive a DW schema from the data sources described by Entity-Relationship (ER) schemas. But we, the authors, observe that this proposal is only oriented towards the conceptual and logical design of the DWs repository, and does not consider important aspects such as the design of ETL processes. Furthermore, Golfarelli at el. assume the existence of all the ER schemas of the data sources, which unfortunately is slightly a rare occurrence. Finally, we, the authors, also observe that the use of a particular notation in [9] can make the practical application of this proposal difficult.

On the other hand, some sincere effort has been made for aligning the design of DWs with the general MDA paradigm, the Model Driven Data Warehouse (MDDW) [10]. This approach is based on the Common Warehouse Metamodel (CWM) [11], which is a platform-independent metamodel definition for interchanging DW specifications between different platforms and tools. Basically, CWM provides a set of metamodels that are comprehensive enough to model an entire DW including data sources, ETL processes, MD modeling, relational implementation of a DW, and so on.

So the authors believe that it is more reliable to design a PIM by using an enriched conceptual modeling approach easy to be handled, and then transform this PIM into a CWM-compliant PSM in order to assure the interchange of DW metadata between different platforms and tools. The Common Warehouse Meta model consists of:

- A common meta model of the data warehousing and business intelligence domain
- A platform-independent meta model definition
- An XML-based inter-change format for metadata
- A mapping to a platform-independent API specification (CORBA IDL)
- Tools that standardize on CWM which can readily share metadata via CWM-compliant XML files

4. MDA Approach to Design a Star Schema

The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a star schema or a snowflake schema. In the literature studied so far, the authors have found that the process of designing the star schema is manual. Hence, in this paper the authors propose an MDA approach to design a star schema for a warehouse which can be fully automated. As an illustration to support the proposition, the authors are taking a simple example of a Result Evaluation System (RES) of an educational institution as an instance to describe the design of data warehouse using star schema. Quick and easy access to data about students' performances is critical to making timely strategic decisions. But access alone isn't enough. Institutes also need to analyze performance data from various perspectives, find patterns, and see the "big picture." With a "big picture view" of individual student's performances, teachers are in a better position to identify potential problems and take the appropriate prescriptive or preventative actions to ensure that each student performs to the best of their abilities.

While designing RES database manually, it is difficult to handle technical complexity persistent in database system due to the problems of maintainability and reusability. Hence, the authors here propose a design for the prototype database for this system using MDA approach. In database design process using this approach, MDA enables the segregation of RES database into various models at different level of abstraction**.** The first level of abstraction as a CIM describes requirements of the system which specifies the business model. At the second level of abstraction, a PIM describes the software specifications defining the domain model of the system. The third level of abstraction as a PSM describes the software realization model and defines detailed design of the system. Then, the resulting database design can be implemented for a particular required platform. Finally, we propose how to derive the star schema for the warehouse design for RES.

4.1 Computation Independent Models (CIM)

In MDA, system requirements are modeled using a Computation Independent Model (CIM). This model is called business model and it uses a vocabulary that is familiar to the domain experts. A CIM does not show details of the systems structure, but the environment in which the system will operate, being useful to understand the problem [12]. Asadi et al. in [13] have defined the scope of the software system through problem domain analysis and also the unambiguous black-box definition of the system, objectives, and scopes has been proposed. After understanding business needs of RES, as an MDA software developer, we specify the following activities that may occur in the appropriate working of RES:

- Basic information of all students is stored in one database file. Basic information will consists of Students' RollNo., Name, FatherName, Class, Semester etc.
- All the information about their scores in the different parameters of evaluation is stored in a separate database file, which will contain Marks_ID, Subject_ID, Markks in Test, Marks in Assignment, Marks in Mid-semester Examinations and Marks in Seminar etc.
- Information about the various subjects studied by the students becomes a part of a separate database file and this will contain attributes like Subject_ID, Subject Name and Teacher_ID for the subject defined.

- Information about the teachers teaching various courses becomes a part of another database file which will contain attributes like Teacher_ID, Teacher_Name and subject_ID of the subject being taught by the respective teacher.

Figure 3 presents a CIM that has been developed for Students Result Evaluation System (RES) and consists of business entities and the relationships among them. After capturing the requirements in CIM, the next step in the process is to generate the Platform Independent Model i.e. The CIM forms the basis of deriving the PIM for the ERS.

4.2 Platform independent models (PIM)

At the second level of abstraction in MDA is defined the Platform Independent Model (PIM). This model is at a relatively lower level of abstraction and by definition, is independent of any implementation technology. This PIM is a platform independent analysis model can be derived by analyzing the requirements model and will form the basis for the design of the DW [14]. The system functionalities are described through this PIM maintaining traceability to the requirements model.

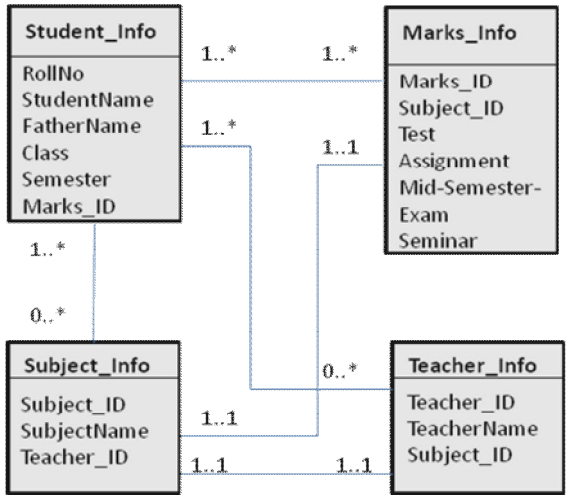

Fig. 3. CIM for RES

Developers may use appropriate model elements stored in a model repository to produce some parts of this PIM. This model is not the final PIM, but forms the foundation for producing the final version. Conventional OO analysis techniques can be used for this activity, which is typically executed in an iterative and incremental manner [13]. PIM aims to capture implementation-independent information about the system and business process modeled.

As an OMG standard, MDA uses the UML models as its core representation. IBM's Rational Rose 2003 and OMG's XML Metadata Interchange (XMI) are thus popular tools that support the process [15]. As shown in figure 4, PIM reflects all the information needed to describe the platform independent system behavior.

The key features of the PIM are as follows [15]:

- High level abstraction that is independent of any implementation technology.
- System is modeled from the viewpoint of how it best supports the business logic.
- Describes system behavior, independent of the computing environment & implementation technologies.
- PIMs can be reused across multiple platforms.

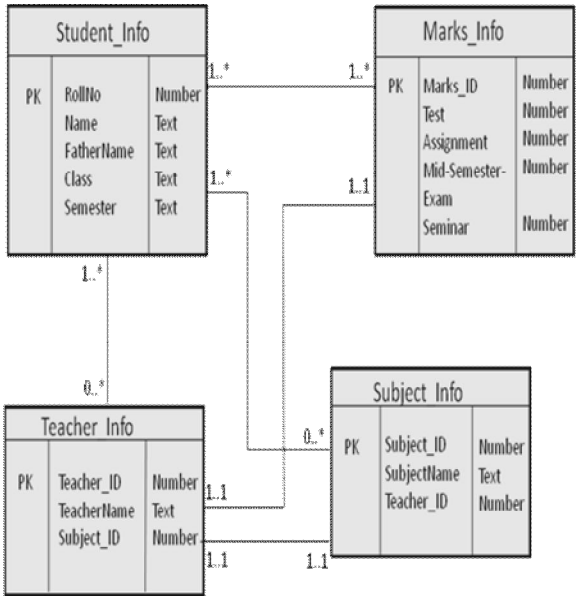

Fig. 4. PIM for RES

5. Star Schema for Design of Data Warehouse for RES

Star Schema is the most common modeling paradigm for multi-dimensional databases, in which the data warehouse contains: (a) fact table, containing the bulk of the data, with no redundancy, and (b) a set of dimension tables, one for each dimension. Figure 5 depicts the star schema of data warehouse for RES using dimension tables like StudentInfo, MarksInfo, SubjectInfo, TeacherInfo. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table [14].

5.1 Platform specific models (PSM)

Once the PSM has been generated [15], next step in the software development process is generating the code from PSM and deploying the system in the specific environment. It may be desirable to support several different deployment configuration e.g. deployment, test and production - each with its own specification of database access paths, user

authentication, authorization and logging mechanisms. Developing formal and automatic transformations among models (e.g. PIM-PSM) is the main advantage of MDA. The declarative approach of QVT for transformation is given in [8], and accordingly relations between elements of the metamodels are used for constructing the transformation between models (i.e. PIM and PSM).

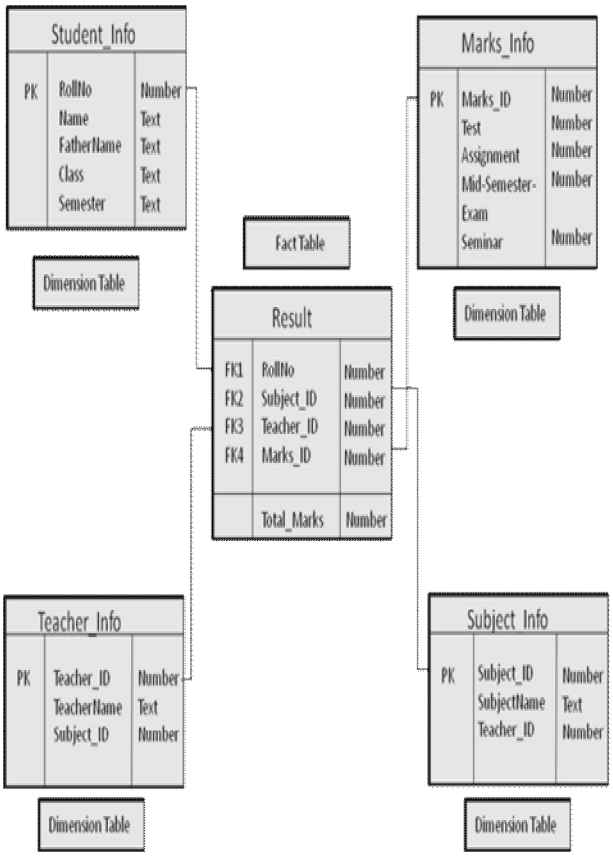

Fig. 5. Star schema for RES

We have used Oracle to define the structure of Dimension tables and fact table.

```
create table Student_Info
(
      rollno number(3) primary key,
      name varchar2(30) not null,
      father_name varchar2(20) not null,
      class varchar2(20) not null,
      semester varchar2(10) not null
)
create table Marks_Info
(
      marks_id number(3) primary key,
      test number(3) not null,
      assignment number(3) not null,
      mid_semester_exam number(3) not null,
      seminar number(3) not null
)
create table Teacher_Info
(
      teacher_id number(3) primary key,
      teacher_name varchar2(30) not null,
      subject_id number(3) not null
)
create table Subject_Info
(
      subject_id number(3) primary key,
      subject_name varchar2(30) not null,
      teacher_id number(3) not null
)
Create table Result
(
      rollno number(3),
       subject_id number(3),
       teacher_id number(3),
       marks_id number(3),
      foreign key rollno references
            student_info(rollno),
      foreign key subject_id references
             subject_info(subject_id),
      foreign key teacher_id references
            teacher_info(teacher_id),
      foreign key marks_id references
            marks_info(marks_id)
)
```

After defining all the structures for the Dimensional Tables and the Fact Table, a complex query based upon the 'Join' is proposed to retrieve the relevant information from the respective sources.

Select SI.rollno, SI.Student_Name, MI.Marks_ID, MI.Test, MI.Assignment,MI.Mid-Semester-Examination, I.Seminar, SI.Subject_ID,SI.Subject_Name,TI.Teacher_ID,I.Teacher_ Name from Student_Info SI, Marks_Info MI,Subject_Info SI, Teacher_Info TI where SI.Marks_ID = MI.Marks_ID and MI.Subject_ID = SI.Subject_ID and SI.Teacher_ID = TI.Teacher_ID group by SI.Marks_ID

6. Conclusion and Future Work

Because of the difficulties to handle technical complexity found in systems, problems of maintainability and reusability persist in database system and also in data warehouses. The MDA software development approach in the recent years has been proved to be able to improve upon these factors by separating the concerns through abstractions at various levels. The schema design of a data warehouse is quite a complex task too. The authors in this paper have suggested that there is no consensus among researchers and practitioners on a common design method for this schema design. Therefore, the authors in this paper have suggested the adopting of MDA approach for the star schema design of a data warehouse.

The main advantage of MDA approach is that once every PIM needed has been developed, one can automatically obtain the PSM and code by using a set of clear and formal transformations, so less time and effort is needed to develop the whole DW. With the help of a suitable illustration of various models of MDA which include CIM, PIM & PSM, it has been demonstrated how the star schema and the corresponding PSM can be generated for a data warehouse. The whole process of schema generation up to the code generation has been shown to include the transformations from CIM to PIM and from PIM to star schema and from star schema to PSM and ultimately to the executable code. The example here has been worked out manually, but the whole of this process can be fully automated with the help of transformation tools. Some of these tools for the part process already exist and the rest can be easily developed by defining the transformation definitions for them. The authors are in the initial stages of developing such a tool for the data warehouse. Also, the example demonstrated in this paper included the generation of a star schema for a simplified PIM, a more complex and practical example needs to be worked out and the authors are also working on one such problem.

References

[1] R. Kimball, The Data Warehouse ETL Toolkit, Wiley Publishing, Inc., 2004.

[2] M. Jarke, M. Lenzerini, Y. Vassiliou, P. Vassiliadis, Fundamentals of Data Warehouses, Springer, 2000.

[3] S. Luján-Mora, J. Trujillo, A Data Warehouse Engineering Process, 3rd Biennial International Conference on Advances in Information Systems (ADVIS'04), Lecture Notes in Computer Science, Izmir, Turkey, Springer-Verlag, 2004.

[4] J. Trujillo, S. Luján-Mora, A UML Based Approach for Modeling ETL Processes in Data Warehouses, 22nd International Conference on Conceptual Modeling (ER'03), Lecture Notes in Computer Science, vol. 2813, pp. 307-320, Chicago, USA, Springer-Verlag, October 2003.

[5] S. Luján-Mora, P. Vassiliadis, J. Trujillo, Data Mapping Diagrams for Data Warehouse Design with UML, 23rd International Conference on Conceptual Modeling (ER'04), Shanghai, China, Lecture Notes in Computer Science, Springer-Verlag, November 2004.

[6] S. Luján-Mora, J. Trujillo, Physical Modeling of Data Warehouses using UML, ACM 7th International Workshop on Data Warehousing and OLAP (DOLAP'04), Washington, D.C., USA, 2004.

[7] Y. Singh, M. Sood, Model Driven Architecture: A Perspective, IEEE International Advance Computing Conference, IACC 2009, India; DOI: 10.1109/IADCC.2009.4809264.

[8] OMG 2nd Revised Submission: MOF 2.0 Query / Views / Transformations, http://www.omg.org/cgi-bin/doc?ad/05-03-02

[9] M. Golfarelli, D. Maio, S. Rizzi, The Dimensional Fact Model: A Conceptual Model for Data Warehouses, International Journal of Cooperative Information Systems, 7(2-3): 215-247, 1998.

[10] J. Mazon, J. Trujillo, M. Serrano, M. Piattini, Applying MDA to the Development of Data Warehouses, ACM 8th International Workshop on Data Warehousing and OLAP (DOLAP'05), November 4–5, 2005, Bremen, Germany, Copyright 2005 ACM 1-59593-162-7/05/0011.

[11] Poole J.D., Model-Driven Architecture: Vision, Standards and Emerging Technologies, ECOOP 2001, Workshop on Metamodeling and Adaptive Object Models, Eotvos Lorand University, Budapest, Hungary, June 18-20, 2001.

[12] J. Miller, J. Mukerji, MDA Guide Version 1.0.1., Object Management Group, 2003.

[13] M. Asadi, M. Ravakhah, R. Ramsin, An MDA-Based System Development Lifecycle, Second Asia International Conference on Modeling & Simulation, IEEE Computer Society, pp. 836-842, 2008.

[14] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann, San Francisco, CA, 2006.

[15] Y. Singh, M. Sood, Models and Transformations in MDA, CICSyN09, First International Conference on Computational Intelligence, Communication Systems and Networks, IEEE Computer Society, pp.253-258, 2009.