Chapter 3
Data Warehousing Meets MDA
A Case Study for Multidimensional Modeling

Jose-Norberto Mazón and Juan Trujillo

Abstract Multidimensional (MD) models are the cornerstone of data warehouse (DW) systems since they allow users to better understand data for decision support, while the performance is improved. MD modeling consists of several phases in the same way as traditional database design: conceptual, logical and physical. In this paper, we argue that designing a conceptual MD model of a DW and deriving its logical representation are very complex, prone to fail, and time consuming tasks. Specifically, two main issues must be considered: (i) the joint analysis of both, information needs of decision makers, and the available operational data sources that will populate the DW, in order to obtain a conceptual MD model, and (ii) the development of formal and automatic transformations between the conceptual and logical design phases. However, no significant effort has been done to take into account these issues in a systematic, well structured and comprehensive development process. To overcome the lack of such process, in this paper, the MD modeling of the DW is aligned with the Model Driven Architecture (MDA) by specifying how to design the different kind of models (CIM, PIM, and PSM), and the required Query/View/Transformation (QVT) transformations between them. To exemplify every part of our approach, a case study is provided throughout the paper.

3.1 Introduction

Data warehouse (DW) systems integrate huge amounts of historical data from operational sources, thus providing useful information for decision makers in an organization. Both practitioners and researchers agree that the development of these systems must be based on the multidimensional (MD) modeling [7, 9], which structures information into facts and dimensions. A fact contains interesting measures (fact attributes) of a business process (sales, deliveries, etc.), whereas a dimen-
sion represents the context for analyzing a fact (product, customer, time, etc.) by means of dimension attributes hierarchically organized\(^1\). Specialized design approaches for MD modeling have been come up to reduce its inherent complexity [8, 29]. Most approaches start defining a conceptual MD model and then, deriving its logical representation. Conceptual design provides mechanisms to specify an implementation-independent MD model either from operational data sources (bottom-up approaches [5, 6, 11, 34]) or from user requirements (top-down approaches[3, 28, 30, 36]), while the logical design aims at deriving a logical model tailored to one specific database technology on the basis of the conceptual model. The implementation of a DW is then based on a set of guidelines and heuristics to derive a logical representation of the conceptual model, thus trying to reduce the semantic gap between them [29]. However, no significant effort has been done to develop an overall approach which

1. combines both a top-down and a bottom-up strategies in an integrated fashion (i.e. considering data sources and information needs of decision makers in early stages of the development), and
2. bridges the semantic gap between conceptual and logical representations, by means of formal and easy-to-use mechanisms, that allow designers to preserve all information captured by advanced conceptual MD models in logical representations, thus avoiding the very tedious, repetitive and prone to fail task of manually applying guidelines.

Considering the above-mentioned drawbacks, in this paper, we describe how to align the MD modeling for DWs with the Model Driven Architecture (MDA) [22] by means of a case study. In concrete, within our MDA-based approach, a conceptual MD model of the DW is developed from user requirements. This initial MD model must be then reconciled with the data sources. The following step is to apply model transformations to automatically obtain several logical models as a basis of the implementation of the DW. In this paper, we describe each element of this approach. First, there is a Computation Independent Model (CIM) for DWs based on the i* framework [37] which contains the main user requirements of the DW. There is also a Platform Independent Model (PIM) based on our UML (Unified Modeling Language) profile [25] for conceptual MD modeling [11]. A set of Query/View/Transformation (QVT) [23] relations for reconciling the PIM with the available operational data sources is also defined. Then, different Platform Specific Models (PSMs) can be developed as logical models depending on the underlying database technology. Different packages of the Common Warehouse Metamodel (CWM) [20, 21] are used to define each PSM. Finally, we have also defined a set of QVT transformation rules in order to obtain the final implementation of the MD model of the DW.

The remainder of this paper is structured as follows. Section 3.2 presents how MDA has been used until now in DW development. Section 3.3 describes our MDA-based approach for MD modeling of DWs. Throughout this section, a case study is

\(^1\) We refer reader to [33] for a further explanation of the different characteristics of MD modeling.