A practical application of our MDD approach for modeling secure XML data warehouses

Belén Vela [a,*], Carlos Blanco [b,1], Eduardo Fernández-Medina [c,2], Esperanza Marcos [a,3]

[a] Rey Juan Carlos University, C/ Tulipán s/n, 28933 Móstoles, Madrid, Spain
[b] University of Cantabria, Av. De los Castros s/n 39071 Santander, Spain
[c] University of Castilla-La Mancha, Paseo de la Universidad, 4-13071 Ciudad Real, Spain

ARTICLE INFO

ABSTRACT

Data warehouses are systems that provide useful information to support the decision making process, thus improving organizations' business processes. These systems integrate heterogeneous sources which are not only limited to their internal business data but also include data from the Web, the latter of which have become increasingly more important in the decision making process in recent years. This has motivated the extensive use of XML in the implementation of data warehouses, in a manner which facilitates data and metadata interchange among the heterogeneous data sources from the Web and the data warehouse. However, the business information that data warehouses manage is crucial and highly sensitive, and must be carefully protected. Security is thus a key issue in the design of data warehouses, regardless of the implementation technology used. It is important to note that the data available on the Web requires particular security considerations which have been specifically tailored to these systems in order to permit their particularities to be captured correctly. Unfortunately, although security issues have been considered in the development of traditional data warehouses, current research lacks approaches with which to consider security when the target platform is based on XML technology.

In order to deal with this situation, in this paper we propose a methodological approach for the model driven development of secure XML data warehouses. We also specify a set of transformation rules that are able to automatically generate not only the corresponding XML structure of the data warehouse from secure conceptual data warehouse models, but also the security rules specified within the data warehouse XML structure, thus allowing both aspects to be implemented simultaneously. We additionally introduce our secure XML DW development approach, in which the secure conceptual DW data model, the PIM, is transformed into a secure XML DW, as a PSM, by applying a set of transformation rules. Our proposal is validated through the practical application of our model driven development approach for Modeling Secure XML Data Warehouses to a case study, which is based on a central Airport DW. We first describe the transformation rules defined, then use a step by step illustration to show how they will be applied to the secure conceptual model of the case study to obtain the Secure XML Data Warehouse, thus demonstrating the benefits of our proposal, and finally we analyze how to achieve the secure implementation into commercial database management systems, providing details of the secure implementation in Oracle XML DB 11 g.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Data Warehouse (DW) systems provide a Multidimensional (MD) view of huge amounts of historical data from heterogeneous operational sources, thus supplying useful and sensitive information which allows decision makers to improve organizations' business processes. 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 in which a fact can be analyzed (product, customer, time, etc.) by means of hierarchically organized dimension attributes.

Traditional DW systems allow business people to acquire useful knowledge from their organization's data by means of a variety of technologies, such as OnLine Analytical Processing (OLAP) or data mining. However, in order to provide richer insights into the dynamics of today's business, it is currently desirable that the data in the organization be combined with data from outside in order

* Corresponding author. Tel.: +34 91 488 70 03.
  E-mail addresses: belen.vela@urjc.es (B. Vela), Carlos.Blanco@unican.es (C. Blanco), Eduardo.FdezMedina@uclm.es (E. Fernández-Medina), esperanza.marcos@urjc.es (E. Marcos).
  [1] Tel.: +34 942 206762.
  [2] Tel.: +34 926 295300.
  [3] Tel.: +34 91 664 74 91.

to complement the company's internal data with value-adding information (e.g., retail prices of products sold by competitors). Since the amount of data available on the Web has been growing rapidly over the last decade, Web data prove to be more and more useful for this purpose. The principal problem with data from the Web is that they are rather heterogeneous and complex. DW systems designers confront this problem by employing XML technologies in order to make use of this data [49]. On the one hand, Web warehousing uses XML as a means to ameliorate the extraction and integration of heterogeneous Web data in the DW. On the other hand, document warehousing requires XML to deal with unstructured data in DW systems [48]. In both cases XML is used to implement the MD model underlying the DW by defining the corresponding design artifacts (facts, dimensions, measures, hierarchies and so on) in order to facilitate the interchange of data and metadata among heterogeneous data sources and the DW system [56]. The design of XML DWs is therefore a cornerstone when it is necessary to use Web data in the decision making process, a situation which is becoming more and more frequent. In fact, as is affirmed by Ravat et al. in a recent paper [53], the definition of design approaches for XML DW, which offer methodological frameworks based on the Model Driven Architecture (MDA), is necessary and is one of the most interesting challenges for the future in the area of DW development.

Furthermore, every design issue should be considered in the development process of an XML DW. More specifically, one of the most important design issues is security, which has, to date, been surprisingly overlooked during the development of XML DWs. Considering that the information managed by DWs is frequently highly sensitive, and sometimes refers to personal data (protected by the law in most countries), DWs should be protected from unauthorized information accesses (whatever the implementation platform is). In fact, a key requirement underlying these recently developed data management systems is a demand for adequate security, along with fine-grained flexible authorization models and access control mechanisms (since DWs deal mainly with read operations). Therefore, rather than considering security once the system has been completely built, we believe that security and privacy measures should be integrated into all layers of the DW design, from the early stages of its development as another relevant requirement, signifying that much more robust, secure and platform independent products will be produced [43,63].

Our intention is to develop secure XML DWs by considering confidentiality issues during the entire development process, from an early development stage to the final implementation. Our proposal has therefore been aligned with an MDA architecture in which security models are embedded and scattered throughout high level system models, which are then transformed until their final implementation according to the MDA strategy. MDA can be used for this purpose, since it shares some similarities with traditional MD modeling methods [54]: i) a requirements modeling phase is initially applied in order to obtain an abstract business model (i.e. a Computational Independent Model, CIM), which represents the information requirements (i.e. functional and security requirements) for the DW; ii) a conceptual design phase is carried out, whose output is a technology independent and expressive conceptual MD model for the DW (i.e. a Platform Independent Model, PIM); iii) a logical design phase aims to obtain a technology-dependent model (i.e. a Platform Specific Model, PSM) from the previously defined conceptual MD model, and iv) this logical model is then the basis for the implementation of the DW. In fact, this paper represents a piece of a complex architecture which we have developed for secure DW design and implementation over the last few years. In [66], we enriched a CIM model which extends the i* framework with security requirements [74]. In [20] we defined an access control and audit model for DW, which is independent of the modeling paradigm, and in [21] we then particularized that model through a UML extension for the conceptual modeling of a secure DW (PIM). Following with the architecture down, in [60] we defined a secure star schema for DW (PSM) as an extension of the Common Warehouse Model (CWM), and we also defined an OLAP model with which to represent multidimensional models of secure DW at the logical level and to obtain the implementation of a secure DW through SQL Server Analysis Services [8]. Finally, we have defined an engineering process for developing secure DWs though our architecture [65], and in [9] we presented an approach for the reverse engineering of secure DWs, starting from the implementation. The contribution of this paper is, therefore, the new PSM metamodel for Secure XML DW which is integrated into our architecture, and the definition of the transformations to obtain the Secure XML Data Warehouse (PSM) starting from a conceptual model (PIM) of a secure DW.

Section 2 presents the background and related works, and this is followed by an overview of Model Driven Development (MDD) and MDA in Section 3. In Section 4 we introduce the secure XML DW development approach. The PIM is the secure conceptual DW data model, which will be semi-automatically transformed into a secure XML DW, as a PSM, by applying a set of transformation rules. In Section 5 we describe the case study to which our proposal was applied. Section 6 describes the transformation rules defined, along with their application to the case study selected. Finally, in Section 8, we put forward our main conclusions and present our future work. We also include an Appendix A with the complete generated XML Schema code.

2. Background and related work

In this section the background and related work is organized according to the following themes: (1) DW modeling; (2) XML DW modeling; (3) security integration into the design process; and (4) security and access control models for DWs.

2.1. Data Warehouse modeling

DW modeling differs from traditional Database modeling and needs specific approaches which manage multidimensional concepts (facts, dimensions, measures, hierarchies, etc.). Various interesting methodologies for DWs exist, which can be classified according to how they define the DW.

Data-driven proposals are based on Inmon's DW definition [28], and focus on the development of the DW repository from data sources by using a top–down approach. Some interesting data-driven methodologies are: that of Cabbibo and Torlone [14], which proposes a UML logical model for OLAP systems but is limited to data sources defined by Entity-Relationship (ER) schemas; that of Golfarelli et al. [24,25], which deals with data sources expressed with star schemas and defines a methodology that includes what-if analysis models with UML, conceptual design by using its own notation (Dimensional Fact Models) and logical modeling; and finally, the Unified Process, which has been adapted for DWs [63] with the proposal of a six-step methodology, although models are not specified for each development stage.

User-driven proposals are based on Kimball's DW definition [32] and consider the users' requirements to develop the DW repository by using a bottom-up approach. These proposals do not define formal methods and do not include the typical development stages (modeling at the business, conceptual, logical and physical levels). Some of these user-driven methodologies are: Kimball's proposal [32], which builds the DW as a combination of the data marts and uses an ER notation for the multidimensional modeling; the Data Warehouse Method [17] which proposes five iterative stages and uses UML to analyze user's requirements but does not deal with the conceptual and logical modeling stages in any depth; and Carneiro and Brayner's