A Methodology for Multilevel Database Design[1]

Eduardo Fernández-Medina and Mario Piattini

Escuela Superior de Informática. University of Castilla-La Mancha.
Paseo de la Universidad 4, 13071, Ciudad Real. (SPAIN). Tel: 34 926 29 53 00
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es http://www.inf-cr.uclm.es

Abstract. In a Connected Society, the importance of database privacy and security increases considerably. Security must be considered as a fundamental requirement in Information Systems (IS) development, and has to be taken into consideration at all stages of the development, instead of being an isolated and marginal requirement considered once the IS is already finished. We propose a methodology for designing secure databases, which allows to design secure databases taking into account constraints regarding sensitive information from the requirements phase.

1. Introduction

The Connected Society forces companies and enterprises to evolve, and to manage information properly in order to achieve their objectives and survive in the digital era. Organizations depend increasingly on IS, which rely upon large databases, and these databases need increasingly more quality and security. Indeed the very survival of the organization depends on the correct management, security and confidentiality of this information (Dhillon and Backhouse, 2000). The protection of databases is a serious requirement, which must be considered carefully, not as an isolated aspect, but as an element present in all stages of the database life cycle, from the requirement analysis to implementation and maintenance. For this purpose, Hall and Chapman (2002) propose different ideas for integrating security in the system development process, but database security is only considered from the cryptographic point of view. As a consequence, we need to commit our efforts to design databases that are more secure.

All the solutions that have been offered up until now are very important, they are partial and isolated, and they do not solve the problem of database protection globally. Moreover, they do not deal with the security problem at the design level. On the other hands, in the traditional database methodologies, security is not considered. To overcome this problem, we propose a methodological approach, which allows us to design databases taking into account confidentiality from the earliest stages of the design.

We have defined a methodology for designing secure databases that extends different UML models and the Object Constraint Language. It also adapts the Unified Process and the methodology for developing secure databases that is proposed in Cas-

[1] Partially supported by DOLMEN (TIC2000-1673-C06-06) and RETISSI project (TIC2001-5023-E) of the Research Projects Subdirection of the Ministry of Science and Technology.

tano et al. (1994). The methodology allows to create conceptual and logical models of multilevel databases, and to implement them by using *Oracle9i Label Security*. There are a few proposals that try to integrate security into the software development process such as the Semantic Data Model for Security (Smith, 1991) and the Multilevel Object Modeling Technique (Marks et al., 1996), but they have only been prototypes. One more recent proposal is UMLSec (Jürgens, 2002) where UML is extended to develop secure systems. This approach is very interesting, but it only deals with IS, whilst databases are not considered.

2. Methodology Overview

This methodology allows us to classify the information according to its confidentiality properties and to which user roles will have access permissions. It is also possible to specify security constraints on that classification. The methodology ends implementing the multilevel database with Oracle*9i* Label Security.

The methodology will be iterative and incremental, driven by use cases, and centered on the architecture. The stages of the methodology are as follows: *Requirements gathering*, *System Analysis*, *Multilevel Relational Logical Design*, and *Specific Logical Design*.

We have extended Rational Rose, to include and manage automatically the security information and constraints in the multilevel database design process.

2.1. Requirements Gathering

The goal of this stage is to collect and represent requirements considering confidentiality aspects. The most important artifact of this stage is the *extended use case model*, which allows us to represent actors and use cases, indicating confidentiality properties of them through stereotypes. The extended use case model introduces the concept of *secure use case* and *authorized actor*. A secure use case is a use case that should be deeply studied from the point of view of security. An authorized actor is an actor that must have special authorizations in order to carry out a particular use case.

The activities of this stage are as follows: *Gathering initial requirements*, *creating the business model and the system glossary*, *looking for actors*, *looking for use cases*, *looking for persistent elements*, *describing use cases*, *analyzing security in actors and in use cases*, *defining priorities in use cases*, *structuring the use case model*, *looking for relationships between use cases*, and *reviewing use cases*. The most important activity is the *analysis of security in actors and use cases*, which consist of studying whether the use cases have confidentiality requirements, and whether the actors need special authorization in order to carry out the related use case.

2.2. System Analysis

The aim of this stage is to build the database conceptual model, considering all the requirements that have been collected in the previous activities. The conceptual model

is composed of the *extended class diagram* and a set of *security constraints* that are expressed through OSCL language (Piattini and Fernández-Medina, 2001).

The extended class diagram allows to specify confidentiality information in classes, attributes and associations, which indicates the conditions that the subjects have to fulfill to access them. On the other hand, the users are also provided with authorization information. The kinds of security information that have been considered in the methodology are *security levels* and *roles of authorized users*. If one security level is assigned to a class, it means that subjects have to be classified in at least the same level to access the information. If a set of roles is assigned to an element, it means that the subjects have to play at least one of those roles to access the element.

The OSCL language allows the specification of security constraints that define the information about security of classes, attributes or associations, depending on a particular condition. For instance, it is possible to define constraints that specifies that the security level of the objects belonging to a class will be more restrictive if the value of one of its attributes have one particular value. The syntax is easy to understand, because this language is based on the well-known Object Constraint Language.

The activities of this stage are as follows: *Architecture analysis*, *use case analysis*, *classes analysis*, *security analysis*, and *package analysis*.

2.3. Multilevel Relational Logical Design

Once a conceptual model has been developed, we can decide which logical database paradigm is interesting in each case. We have considered relational databases because they are the most used and widespread at present (Leavitt, 2000), but it could be possible to develop secure object-relational or object oriented databases. This stage is the bridge between the conceptual model and the implementation with a specific logical model. The components of the multilevel relational model are as follows: *Database relational model* (includes the definition of each relation or the database, considering the necessary attributes for representing the confidentiality information), *meta-information of the model* (each relation has associated a meta-information tuple, which includes the data type of the attributes, and the valid values of the attributes related to security information), and *security constraints*. The main activities of this stage deal with the transformation of all elements in the extended class diagram into the multilevel relational model.

2.4. Specific Logical Design

At the end of the previous stage, we have built a logical model of the secure database that is implementation independent. In this stage we specify the secure database in a particular logical model: Oracle*9i* Label Security. We have chosen this model because it is part of one of the most important DBMS that allows the implementation of label-based databases.

The activities of this stage are as follows: *Defining the database model* (all the tables), *defining the security policy and their default options*, *defining the security information in the security policy*, *creating the authorized users and assigning their au-*

thorizations, defining security information for tables through labeling functions, implementing security constraints through labeling functions and access control predicates, and finally, *implementing operations and controlling their security*.

3. Conclusions

Confluence of databases, IS and their application in business, together with new requirements of laws and governments, make necessary more sophisticated approaches to ensure data security.

Traditionally, information security deals with different research topics, like access control techniques, cryptographic methods, etc. Although all these topics are very important, we think that we should use a methodological approach, where security, at different levels, is taken into account at all stages of the database development process. In this paper we have summarized a methodology that has been created thinking about security and usability, extending the modeling languages, process models, constraint languages and security models that are most accepted in the industrial and research community. We have used the methodology to design a secure database in a Spanish local government, and we have solved its confidentiality problems.

4. References

Castano, S., Fugini, M., Martella, G. and Samarati, P. (1994). Database Security. Addison-Wesley.

Dhillon, G. & Backhouse, J. (2000). Information system security management in the new millennium. *Communications of the ACM,. 43, 7,* 125-128.

Hall, A. & Chapman, R. (2002). Correctness by construction developing a commercial secure system. *IEEE Software, 19, 1,* 18-25.

Jürjens, J. (2002). UMLsec: Extending UML for secure systems development. In Jézéquel, J., Hussmann, H. & Cook, S. (Eds.), *UML 2002 – The Unified Modeling Language, Model engineering, concepts and tools* (pp. 412-425). Germany, Springer.

Leavitt, N. (2000). Whatever happened to Object-Oriented Databases?. Industry Trends, IEEE Computer Society, August, 16-19.

Marks, D., Sell, P. and Thuraisingham, B. (1996). MOMT: A Multilevel Object Modeling Technique for Designing Secure Database Applications. Journal of Object-Oriented Programming. Vol. 9. Nº 4, pp. 22-29.

Piattini, M. & Fernández-Medina, E. (2001). Specification of security constraints in UML. In proceedings of the 35th Annual 2001 IEEE International Carnahan Conference on Security Technology (ICCST 2001), pp. 163-171. October, 2001. London (UK).

Smith, G.W. (1991). Modeling Security-Relevant Data Semantics. Proceedings of the IEEE Transactions on Software Engineering, Vol. 17. Nº 11, November pp. 1195-1203.