

Tool Support for Enforcing Security Policies on Databases

Jenny Abramov², Omer Anson², Arnon Sturm¹, Peretz Shoval¹

¹Department of Information Systems Engineering

²Deutsche Telekom Laboratories (T-Labs)

Ben-Gurion University of the Negev

Beer Sheva 84105, Israel

jennyab@bgu.ac.il, oaanson@gmail.com,
sturm@bgu.ac.il, shoval@bgu.ac.il

Abstract. Security in general and database protection from unauthorized access in particular, are crucial for organizations. It has long been accepted that security requirements should be considered from the early stages of the development. However, such requirements tend to be neglected or dealt-with only at the end of the development process. The Security Modeling Tool presented in this study aims at enforcing developers, in particular database designers, to deal with database authorization requirements from the early stages of development. This software demonstration shows how the Security Modeling Tool assists to define organizational security policies and use them during the application development to create a secured database schema.

Keywords: Secure software engineering, database design, authorization.

1 Introduction

Data is the most valuable asset for an organization as its survival depends on the correct management, security, and confidentiality of the data [1]. In order to protect the data, organizations must secure data processing, transmission and storage. Developers of data-oriented systems always face problems related to security. This is the case as security and other non-functional requirements are usually ignored in the early stages of the development process.

To overcome these lacks, we provide a methodology that guides developers in the incorporation of particular organizational security policies, as well as verifying their correct application. In addition, the methodology enables the developer to transform the result into code.

The methodology incorporates ideas from two areas of expertise: in the area of *methodologies for system development*, we adopt the principle of integrating data and functional modeling at the early stages of the development, suggested by the Functional Object-Oriented Methodology (FOOM) [6]. Additionally, in the area of *domain engineering*, we adopt the principles suggested by the Application Based Domain Modeling (ADOM) approach [4]. ADOM supports building reusable assets on the one hand, and representing and managing knowledge in specific domains on

After creating a refined data model, we need to check if it adheres to the security policies as defined by the specified security patterns. The SMT provides automatic verification that is based on the ADOM validation algorithm. If the application is invalid, an error message is presented, explaining the verification errors. An example of two such errors are: 1) multiplicity error: access type is not specified to the *Privilege* class *StudentR_CourseOffering*, and 2) OCL error: *StudentR* role has the *SYSDBA* privilege.

The Implementation Phase. During this phase the transformation rules, which were defined during the preparation phase, are used to translate the verified application model into a database schema.

4 Summary

We have presented the Security Modeling Tool, which supports the development of secured database schemata upon the methodology we have developed. This tool utilizes security patterns for enforcing security on database application design. The tool guides developers on how to incorporate security aspects, in particular authorization, within the development process with pre-defined security patterns. It handles the specification and implementation of the authorization aspect from the early stages of the development process, leading to a more secure system design.

Currently, we are in a process of applying the methodology along with its supporting tool within an industrial environment. This will enable us to introduce improvements in the methodology and the tool. In future work, we plan to enrich the methodology and tool to support other non-functional requirements (e.g., in the security era it might include privacy, encryption, and auditing). In addition, we plan to apply the methodology to the code level, similarly to the way we apply it to database schemata; yet, we intend to incorporate behavioral specification as well.

References

1. Dhillon GS. Information Security Management: Global Challenges in the New Millennium. IGI Publishing (2001)
2. Eclipse Modeling Framework (2011). <http://www.eclipse.org/modeling/emf/>
3. Jouault F, Allilaire F, Bézivin J, Kurtev I. ATL: A model transformation tool. *Science of Computer Programming*. 72(1-2), 31--39 (2008)
4. Reinhartz-Berger, I., Sturm, A.: Utilizing Domain Models for Application Design and Validation. *Information & Software Technology* 51 (8), 1275--1289 (2009)
5. Schumacher, M.: *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*. Springer-Verlag New York, Inc., Secaucus (2003)
6. Shoval, P.: *Functional and Object-Oriented Analysis and Design - An Integrated Methodology*. IGI Publishing, Hershey (2007)
7. Standard Widget Toolkit (2011). <http://www.eclipse.org/swt/>
8. StringTemplate (2011). <http://www.stringtemplate.org/>