Chapter 5
Integrating Access Control into UML for Secure Software Modeling and Analysis

Thuong Doan
University of Connecticut, USA

Steven Demurjian
University of Connecticut, USA

Laurent Michel
University of Connecticut, USA

Solomon Berhe
University of Connecticut, USA

ABSTRACT

Access control models are often an orthogonal activity when designing, implementing, and deploying software applications. Role-based access control (RBAC) which targets privileges based on responsibilities within an application and mandatory access control (MAC) that emphasizes the protection of information via security tags are two dominant approaches in this regard. The integration of access control into software modeling and analysis is often loose and significantly lacking, particularly when security is such a high-priority concern in applications. This paper presents an approach to integrate RBAC and MAC into use-case, class, and sequence diagrams of the unified modeling language (UML), providing a cohesive approach to secure software modeling that elevates security to a first-class citizen in the process. To insure that a UML design with security does not violate RBAC or MAC requirements, design-time analysis checks security constraints whenever a new UML element is added or an existing UML element is modified, while post-design analysis checks security constraints across the entire design for conflicts and inconsistencies. These access control extensions and security analyses have been prototyped within a UML tool.

DOI: 10.4018/978-1-4666-1580-9.ch005

INTRODUCTION

The inclusion of security in software design and development has often been an afterthought, delayed to near or post-deployment stages of the software development process or delegated to database administration. However, security has emerged as fundamental concern early and in all phases of the software process, prevalent in user interfaces (to control what each user can see), functional capabilities (to control what each user can do), and repositories (to control what data a user can access/modify). The specific focus of the paper is the integration of access control into the UML (Booch et al., 1999), to provide the means for software designers and engineers to jointly model their application's functional and security requirements and constraints, augmented with analyses that insures the access control model characteristics, capabilities, and constraints that are being utilized are not violated. Note that despite the existence of parallels between security and elements in UML, direct support for security specification is not provided (OMG).

For access control, we leverage: role-based access control (RBAC) that focuses on user responsibilities via roles (Sandhu et al., 1996; Ting, 1988) with constraints to restrict behavior (Ferraiolo et al., 2001); and, mandatory access control (MAC) that defines classifications for objects and clearances for subjects (Bell & La Padula, 1975) with access based on the relationship between subjects and objects (Biba, 1977; Osborn et al., 2000). For security, RBAC is a flexible approach to grant/revoke permissions to/from users via roles, while MAC controls information flow (read/write on objects) for highly secure systems. Both models are augmented in this approach with lifetime constraints that determine a temporal window of activity for privileges.

This paper details a practical approach that integrates RBAC and MAC into UML for secure software modeling and analysis with a two-fold emphasis. First, UML requirements definition (use case diagram) and design (class and sequence diagrams) are extended with visual and non-visual security capabilities and constraints for MAC, lifetime, and RBAC. Second, security modeling is augmented with analyses via the checking of security constraints. The *design-time analysis* checks these constraints as the application is created and changed as a result of every action taken by an engineering/designer. The *post-design analysis* (akin to a compile) checks these constraints across the entire application at a particular increment. Both the modeling and analysis capabilities have been programmatically integrated into Borland's UML design tool Together Architect (2009). The snapshots of the implementation illustrate the examples in the paper. This work goes beyond our prior efforts (Doan et al., 2004a; Doan et al., 2004b) by collectively bringing all of the security extensions to UML along with their respective analyses into one context that clearly demonstrates the capabilities and potential of the work in total.

The work presented herein contrasts with other efforts on security for UML. The work of Shin and Ahn (2000) and Ray et al. (2003) simply *uses* UML to represent MAC and/or RBAC systems, as opposed to explicitly *extending* UML with RBAC and MAC. UMLsec (Jurjens, 2002a, 2002b) focuses on multi-level security (MAC) of message in sequence/state diagrams and is similar to our work on MAC extension. SecureUML (Lodderstedt et al., 2002) introduces new meta-model components and authorization constraints expressed for RBAC that involve meta-model changes. Lastly, Alghathbar and Wijesekera (2003a) incorporate security into use cases, similar to our approach. The main difference between our approach and others is one of comprehensiveness; we are doing RBAC, MAC, constraints, and lifetimes for temporal access, which combines many features of the aforementioned work with other capabilities that they do not provide.

This remainder of this paper is organized as follows. Section 2 provides brief background on UML and access control models. Section 3 intro-