Chapter 7.9
Access Control Specification in UML

Manuel Koch
Free University of Berlin, Germany

Francesco Parisi-Presicce
George Mason University, USA and University of Rome "La Sapienza", Italy

Karl Pauls
Free University of Berlin, Germany

ABSTRACT

Security requirements have become an integral part of most modern soft¬ware systems. In order to produce secure systems, it is necessary to provide soft¬ware engineers with the appropriate systematic support. This chapter discusses a methodology to integrate the specification of access control policies into UML. The methodology, along with the graph-based formal semantics for the UML access control specification, allows to reason about the coherence of the access control spec¬ification. The chapter also presents a procedure to modify policy rules to guarantee the satisfaction of constraints, and shows how to generate access control requirements from UML diagrams. The main concepts in the UML access control specification are illustrated with an example access control model for distributed object systems.

INTRODUCTION

Security requirements are an important aspect in the development of software systems that are not used in completely trusted environments. In order to increase the overall system security and to better satisfy security constraints, security policies should be specified in terms of security models that are integrated with the general software engineering models. Modelling techniques that are well known to software engineers should be used, to prevent them from specifying mistakes in security

specifications due to inadequate expertise of the particular specification technique. The reasoning about and the verification of security properties require a formal semantics for the specification technique.

Since the UML is nowadays the de-facto standard modelling language, the usage of UML for the specification of security aspects is particularly attractive, since software engineers are used to the UML notation and the accompanying tools (Brose, Koch, & Lohr, 2002; Devanbu & Stubblebine, 2000; Epstein & Sandhu, 1999; Jurjens, 2001; Lodderstedt, Basin, & Doser, 2002). We identify in this chapter the parts necessary to specify an access control policy and propose a UML specification for these parts. In our framework, we use only existing UML model elements and extension mechanisms to ensure compatibility with UML tools that can then be directly used for the access control specification. The UML specification of an access control model makes use of UML class diagrams, object diagrams, some additional stereotypes and OCL constraints (OMG, 2003).

Access control constraints restrict an access control model and prevent the system from reaching unwanted protection states. The expression and specification of access control constraints is a difficult task and existing languages are often too complex for administrators to determine whether a set of constraints really satisfies the requirement. Therefore, access control languages have been proposed which have a manageable complexity and are understandable by administrators but still expressive enough to capture most practical access control constraints (Ahn & Sandhu, 2001; Jaeger & Tidswell, 2001; Koch, Mancini, & Parisi-Presicce, 2002c). We present in this chapter an approach to specify access control constraints in UML. We consider OCL constraints as a textual presentation of access control constraints, but present also visual constraint specification by UML object diagrams (Koch et al., 2002c; Ray, Li, France, & Kim, 2004).

In our approach, access control models are specified on several levels, depending on the added application domain information similar to the UML meta-modelling architecture (without the metametamodel). On the metamodel level, there is no application domain information and the access control specification shows only the policy rules and constraints. This is the level at which the generic access control model, such as Role-Based, Mandatory, Discretionary, are described. The model level is an instance of the meta access control model and refines it using application domain information. At this level, the specific roles used in RBAC or the clearance, and categories used to construct the labels in MAC are defined depending on the application. The last level is an instance of the access control model in a specific information domain. This is the level at which principals are assigned to roles or labels, or specific permissions are assigned to roles.

To verify that an access control policy satisfies all the access control constraints (i.e., the policy permits only system states that satisfy all the constraints) an appropriate formal semantics for the UML access control model is needed. We give a graph-based formal semantics (Rozenberg, 1997) to the UML access control specification by a transformation of UML class and object diagrams, respectively, into graphs and graph rules. The graph-based semantics enables us to use several verification concepts to verify the constraints with respect to the access control model (Koch, Mancini, & Parisi-Presicce, 2001; Koch, Mancini, & Parisi-Presicce, 2002b).

To know how to model access control requirements into UML is only part of the problem. Another essential part is to determine the access control requirements themselves. We present an approach to generate automatically access control requirements from UML use case, class, and sequence diagrams. The extracted access control information is the basis for the UML access control model.