Chapter III

Enterprise Modeling with the Joint Use of User Requirements Notation and UML

Anna Medve, University of Vezprém, Hungary

ABSTRACT

This chapter introduces the user requirements notation (URN) standardized formal methods and its joint use with unified modeling language (UML) in enterprise modeling. It argues that the joint use of URN and UML can enhance enterprise models and coevolve with enterprise engineering processes. URN combines goals and scenarios to help reasoning and to capture user requirements prior to detailed design. Furthermore, URN can be integrated partially or entirely into an existing business process modeling approach without replacing current ways of creating and analyzing models in order to be useful. Modeled in the UML, a URN model may be incorporated into the rest of a system’s UML design documentation, seamlessly linking the documentation for the requirements elicitation part of a project to the whole, and to be fully integrated with the rest of the design documentation for a software system.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
BACKGROUND

Enterprise roles must aid in the design of information technology (I.T) infrastructure and its interfaces between software systems that adapt enterprise organizational structure to maintain a consistent overview of all processes (ATHENA, 2004). Enterprise roles are involved in modeling processes via their facility to help designers to link the modeling, optimization, and execution of business processes with their physical configurations, IT, and software systems. In order to have a common language that is understandable by all participants, the modeling language should be a common language between the stakeholders and the system designers.

Two aspects important in object orientation in modeling practices must be highlighted: The first aspect is that the object management group (OMG) defines object management as software development that models the real-world through representation of "objects." These objects are the encapsulation of the attributes, relationships, and methods of software-identifiable program components. Object management results in faster application development, easier maintenance, enormous scalability, and reusable software (OMG, 2003).

The second aspect is that open distributed processing–reference model (ODP-RM) is a standard elaborated by the joint working group ISO/IEC/ITU. ODP supports the modeling of distributed processing entities using UML metamodel. The term "object" in ODP means "a model of entity." An entity is any concrete or abstract thing of interest. An object

Figure 1. UCMs as a missing piece of the UML puzzle (Amyot, He, He, & Cho, 1999)