Chapter 7
Formal Consistency Verification of UML Requirement and Analysis Models

Mouez Ali
University of Sfax, Tunisia

Hanene Ben-Abdallah
University of Sfax, Tunisia

Faïez Gargouri
University of Sfax, Tunisia

ABSTRACT

To capture and analyze the functional requirements of an information system, UML and the Unified Process (UP) propose the use case and sequence diagrams. However, one of the main difficulties behind the use of UML is how to ensure the consistency of the various diagrams used to model different views of the same system. In this chapter, the authors propose an enriched format for documenting UML2.0 use cases. This format facilitates consistency verification of the functional requirements with respect to the sequence diagrams included in the analysis model. The consistency verification relies on a set of rules to check the correspondence among the elements of the documented use cases and those of the sequence diagrams; the correspondence exploits the implicit semantic relationship between these diagrams as defined in UP. Furthermore, to provide for a rigorous verification, the authors formalize both types of diagrams and their correspondence rules in the formal notation Z. The formal version of the analysis model is then verified through the theorem prover Z/EVES to ensure its consistency.

DOI: 10.4018/978-1-4666-4369-7.ch007

INTRODUCTION

Requirement engineering (RE) is the front-end activity in software development. It includes requirement elicitation/capturing followed by requirement analysis. Being a *de facto* standard, several development processes have been proposed to derive requirement models using UML (e.g., Rational Unified Process [RUP] [Kruchten, 1999] and the Unified Process [UP] [Jacobson et al., 1999]). According to UP, the functional requirements can be modeled with a use case diagram in three phases. In the first phase, the designer identifies all the *actors* or organizations related to the system under development. In the second phase, he/she identifies the set of functions pertinent to each actor; these functions are represented by *use cases*. Finally, in the third phase, the designer must document the use cases using a natural language.

Once captured, the requirements are then analyzed by the designer through a set of analysis models. With UP, the designer analyzes the use case diagram by specifying UML sequence and/or collaboration diagrams. In this UP step, the designer identifies a set of *objects* and *messages* to illustrate the interactions among the actors and the system's components.

Evidently, both the requirement and analysis models are semantically related and complementary in presenting various details about the system. Thus, a basic hypothesis is the consistency of the analysis models with respect to the captured requirements. In addition, any consistency verification approach must explicitly take into account both the syntactic and semantic dependences between the two models. The syntactic relationships between the requirements and analysis models can be derived through the UML meta-model (OMG, 2003). On the other hand, UP informally specifies the semantic relationships between these two models: UP considers that the use case diagram is *specified by* the sequence diagrams and that these latter are *equivalent* to the collaboration diagrams. However, being informally specified through derivation "good-practice" guidelines, these two semantic relationships do not confidently ensure the consistency of these models.

This UP limitation motivated several works (*cf.* (Engels et al., 2001; Reggio et al., 2001; Yang et al., 2004; Liu & Araki, 2005) to propose consistency verification approaches for UML models. The efficiency of the proposed approaches is limited by the high level of abstraction of the use cases. Furthermore, the use case documentation does not resolve this difficulty since the documentation formats so far used are unstructured and are expressed in natural language.

The first contribution of this chapter is the proposition of a new format for use case documentation that facilitates the specification of the set of interactions between the actors and the system. Its second contribution is the exploitation of this documentation to *formally* verify the consistency of the requirements and analysis models. For this, we formalize in Z (Spivey, 1992) the meta-models of the use case

Related Content