Chapter IX

Formal Specification of Software Model Evolution Using Contracts

Claudia Pons, Universidad Nacional de La Plata, Argentina

Gabriel Baum, Universidad Nacional de La Plata, Argentina

Abstract

During the object-oriented software development process, a variety of models of the system is built. All these models are semantically overlapping and together represent the system as a whole. In this chapter, we present a classification of relationships between models along three different dimensions, proposing a formal description of them in terms of mathematical contracts, where the software development process is seen as involving a number of agents (the development team and the software artifacts) carrying out actions with the goal of building a software system that meets the user requirements. In this way, contracts can be used to reason about correctness of the development process, and to compare the capabilities of various groupings of agents in order to accomplish a particular contract. The goal of the proposed formalization is to provide formal foundations for tools that perform intelligent analysis on models assisting software engineers through the software life cycle.
Introduction

A software development process is a set of activities that jointly convert users’ needs to a software system. Modern software development processes, such as the Unified Process (Jacobson, Booch, & Rumbaugh, 1999), are iterative and incremental, they repeat over a series of iterations making up the life cycle of a system. Each iteration takes place over time and consists of one pass through the requirements, analysis, design, implementation, and test activities, building a number of different models. Due to the incremental nature of the process, each iteration results in an increment of models built in previous iterations. This creates a natural relationship between the elements among different phases and iterations; elements in one model can be related to elements in another model. For instance, a use case (in the use case model) can be traced to a collaboration (in the analysis or design model) representing its realization. Figure 1 lists the classical phases or activities — requirements, analysis, design, implementation, and test — in the vertical axis and the iteration in the horizontal axis. Three different dimensions are distinguished in order to classify relationships between models:

- horizontal dimension (internal dimension)
- vertical dimension (activity dimension)
- evolution dimension (iteration dimension)

The horizontal dimension deals with relations between submodels that coexist consistently making up a more complex model. The UML incorporates several sublanguages, each one allowing a specific view on the system. Models of different viewpoints have a certain overlap, for instance, an analysis model consists of sequence diagrams and

Figure 1. Dimensions in the development process