Chapter II

Using a Graph Transformation System to Improve the Quality Characteristics of UML-RT Specifications

Lars Grunske, University of Potsdam, Germany

Abstract

This chapter presents the concept of graph-based architecture evolution and how this concept can be applied to improve the quality characteristics of a software system. For this purpose, the UML-RT used as an architectural specification language is mapped to a hypergraph-based data structure. Thus, transformation operators can be specified as hypergraph transformation rules and applied automatically.

Introduction

Over the past few years, software intensive technical or embedded systems have increasingly been implemented in software components (Douglas, 1999; Gomaa, 2000; Liggesmeyer, 2000). These software components have to fulfill requirements relating to quality characteristics or nonfunctional properties (NFPs), such as safety, availability, reliability, and temporal correctness. If a system does not fulfill these requirements, the
system must be restructured to improve the quality characteristics. Due to economical reasons, this change must be made as early as possible, preferably in the design phase, after the development of the system/software architecture. Based on the architecture specification, for the first time, an evaluation of the quality characteristics of the system is possible.

For the restructuring of software architectures in Bosch and Molin (1999), a cyclic process (see Figure 1) is presented that can be used to improve the quality characteristics of software architectures. The precondition for its application is an architectural specification that fulfills all functional requirements. Based on this specification, the quality characteristics are determined by an evaluation of the architecture. If the architectural specification does not meet its quality requirements, the software architecture must be restructured by the application of transformation operators. These transformation operators should influence the quality characteristics without changing the functional behavior. Thus, after the transformation the architectural specification is still functionally correct. If it turns out that all quality characteristics meet their corresponding requirements, the cyclic process can be terminated and system development can proceed with the detailed design and the implementation phase.

This chapter presents the concept of hypergraph-based architectural evolution and how this concept can be applied in the process model. For this purpose UML-RT is used as an architectural description language and the relevant elements of the UML-RT metamodel are mapped to a hypergraph-based data structure. The main benefit of this approach is the possibility to model architecture transformations as hypergraph transformation rules. Consequently, this approach allows for a (semi-) automatic application. Due to the complexity of the overall setup and the precision needed, it becomes inevitable to support the evolution process with an appropriate utility. For this purpose, a tool called Balance has been developed, which provides facilities for applying the architectural transformations explained previously.

To clarify the understanding of the hypergraph-based architecture evolution, we are going to explain these items more precisely in the following sections. In the second

Figure 1. Cyclic process for the improvement of quality characteristics