Chapter 2.30
Evaluating Performance of Software Architecture Models with the Palladio Component Model

Heiko Koziolek
Universität Oldenburg, Germany

Steffen Becker
University of Karlsruhe, Germany

Ralf Reussner
University of Karlsruhe, Germany

Jens Happe
Universität Oldenburg, Germany

ABSTRACT

Techniques from model-driven software development are useful to analyse the performance of a software architecture during early development stages. Design models of software models can be transformed into analytical or simulation models, which enable analyzing the response times, throughput, and resource utilization of a system before starting the implementation. This chapter provides an overview of the Palladio Component Model (PCM), a special modeling language targeted at model-driven performance predictions. The PCM is accompanied by several model transformations, which derive stochastic process algebra, queuing network models, or Java source code from a software design model. Software architects can use the results of the analytical models to evaluate the feasibility of performance requirements, identify performance bottlenecks, and support architectural design decisions quantitatively. The chapter provides

a case study with a component-based software architecture to illustrate the performance prediction process.

INTRODUCTION

To ensure the quality of a software model, developers need not only to check its functional properties, but also assure that extra-functional requirements of the system can be fulfilled in an implementation of the model. Extra-functional properties include performance, reliability, availability, security, safety, maintainability, portability, etc. Like functional correctness, these properties need to be addressed already during early development stages at the model level to avoid possible later costs for redesign and reimplementation.

Performance (i.e., response time, throughput, and resource utilization) is an extra-functional property critical for many business information systems. Web-based information systems rely on fast response times and must be capable of serving thousands of users in a short time span due to the competitive nature of internet businesses. Furthermore, the responsiveness of software used within companies is important to ensure efficient business processes.

Performance problems in large distributed systems can sometimes not be solved by adding more servers with improved hardware ("kill it with iron"). Large software architectures often do not scale linearly with the available resources, but instead include performance bottlenecks that limit the impact of additional hardware.

Therefore, it is necessary to design a software architecture carefully and analyse performance issues as early as possible. However, in the software industry, performance investigations of software systems are often deferred until an implementation of the system has been build and measurements can be conducted ("fix it later"). To avoid this approach, which might lead to expensive redesigns, software architects can use performance models for early, pre-implementation performance analysis of their architectures.

This chapter provides an overview of the Palladio Component Model (PCM), a domain specific modelling language for component-based software architectures, which is specifically tuned to enable early life-cycle performance predictions. Different developer roles can use the PCM to model the software design and its targeted resource environment. The models can be fed into performance analysis tools to derive the performance of different usage scenarios. Software architects can use this information to revise their architectures and quantitatively support their design decisions at the architectural level.

The chapter is structured as follows: Section 2 provides background and describes related work in the area of model-driven performance prediction. Section 3 introduces different developer roles and a process model for model-driven performance predictions. Section 4 gives an overview of the PCM with several artificial model examples, before Section 5 briefly surveys different model transformations to analysis models and source code. Section 6 describes the performance prediction for an example component-based software architecture and discusses the value of the results for a software architect. For researchers interested working in the area of model-driven performance prediction, Section 7 highlights some directions for future research. Section 8 concludes the chapter.

BACKGROUND AND RELATED WORK

Model-driven performance predictions aim at improving the quality of software architectures during early development stages (Smith et al., (2002)). Software architects use models of such prediction approaches to evaluate the response time, throughput, or resource utilization to be expected after implementing their envisioned