ABSTRACT

Service engineering and service-oriented architecture as an integration and platform technology is a recent approach to software systems integration. Quality aspects ranging from interoperability to maintainability to performance are of central importance for the integration of heterogeneous, distributed service-based systems. Architecture models can substantially influence quality attributes of the implemented software systems. Besides the benefits of explicit architectures on maintainability and reuse, architectural constraints such as styles, reference architectures and architectural patterns can influence observable software properties such as performance. Empirical performance evaluation is a process of measuring and evaluating the performance of implemented software. We present an approach for addressing the quality of services and service-based systems at the model-level in the context of model-driven service engineering. The focus on architecture-level models is a consequence of the black-box character of services.
INTRODUCTION

With software services becoming a strategic capability for the software sector, a service engineering discipline needs to address service development problems based on suitably flexible modelling and composition support. An increasing need for flexibility in this area is caused by changing user requirements, evolving services, and varying deployment contexts. Software services are applications that are provided ‘as-is’ at certain locations in order to be integrated into existing applications or composed to larger systems. Essential in this process are abstract descriptions or models of the service functionality and other service characteristics. This makes model-driven software development both a highly suitable, but actually also necessary framework to adequately develop service-based software systems. Composition and integration-oriented modelling has already been successfully utilised for model-driven service development. The high complexity of modern software makes its development costly and error-prone. Model-driven development (MDD) is an approach that deals with software complexity by making software models primary artefacts of the software development process. MDD utilises two aspects of models. Firstly, in various engineering disciplines, predictions about a software system can be made based on a model. Secondly, even complete implementations for different platforms and languages can be generated from models.

The aim of this chapter is to address quality aspects of model-driven service engineering. We address two specific facets of quality assurance for model-driven software development: the model-driven design and development of high-quality software and the identification of quality aspects in model-driven development. Some specific aspects that we discuss in the context of quality aspects in model-driven service development are:

- modelling and architecture are concept that are closely linked in the context of model-driven development and service-oriented architecture,
- patterns for the model-driven development of service architectures to structure models and to enhance the integration task,
- model-based design and analysis of specific quality aspects for service-based systems.

We aim to demonstrate that model-driven architecture and design of this specific type of service-based software systems can yield high-quality software. Based on an analysis of the state of the art of service engineering and its platform and application requirements supported by a case study analysis, we identify the central quality aspect pertinent to services. This analysis is necessary in order to justify techniques for the quality-aware model-driven service engineering. Factors that impact the quality are:

- network and platform characteristic impacting on for instance performance,
- service-orientation to enhance interoperability and reusability,
- evolution and change as inevitable factors impacting on maintainability.

Service engineering and service-oriented architecture as an integration and platform technology is a recent approach to service-based software systems integration. Quality aspects, however, have not been addressed in sufficient depth in this context. Our technical contribution is an architecture- and pattern-based model-driven service engineering framework that aims at high-quality models as well as quality implementations. While functionally oriented design patterns have been widely used to support the development of large-scale software systems, we combine a service-specific range of these patterns with distribution patterns, which directly impact service-specific quality aspects such as performance. We complement this architectural perspective with an empirical, model-driven performance evaluation