Chapter 3

Modeling and Analyzing Non-Functional Requirements in Service Oriented Architecture with the User Requirements Notation

Hanane Becha  
University of Ottawa, Canada

Gunter Mussbacher  
University of Ottawa, Canada

Daniel Amyot  
University of Ottawa, Canada

ABSTRACT

Non-functional properties (NFPs) represent an important facet of service descriptions, especially in a Service Oriented Architecture. Yet, they are seldom explicitly described, and their use in service selection and composition is still limited. This chapter presents the User Requirements Notation (URN) as a means to model and analyze functional and non-functional service requirements. Aspect-oriented extensions to URN (AoURN) enable the modeling and modularization of different concerns, including non-functional requirements, which can crosscut services or service components. The chapter also proposes a taxonomy of NFPs used to annotate services and service compositions modeled with AoURN. These annotations enable the specification of quantitative non-functional values for services, guide service selection, and support the computation of the NFP (e.g., the quality of service) of their composition. This approach is illustrated with a simple yet realistic composite service (BookItWell), with an emphasis on four types of NFPs, namely service cost, response time, reliability, and availability.

DOI: 10.4018/978-1-60566-794-2.ch003
INTRODUCTION

In a Service Oriented Architecture (SOA), applications are designed as business processes and realized using services and service composition. A service is a mechanism that enables access to one or more capabilities offered by a provider for use by one or many consumers. Service providers and service consumers may be part of different organizations. Service composition is the capability of combining loosely-coupled (atomic or composite) services, residing in the network and accessible via standardized protocols, into a larger composite service by defining when and under which conditions the composite service invokes the services of which it is built. If concrete services are selected at design time, then the composition is called static composition. Otherwise, the composition is dynamic and is based on abstract services until runtime, at which point concrete services are selected. An abstract service is only a description of the functionality of the service and is not associated with any specific concrete service. Due to the dynamic nature of the SOA environment, as services are replaced or upgraded and new services are developed, dynamic composition becomes more effective than static composition. It enables business processes to be implemented from the latest existing services and those implementations to be adjusted quickly to meet changing business requirements and an ever evolving context.

Although many such requirements target the functionalities of desired services, it is also essential to consider the non-functional requirements in order to satisfy the relevant goals of the parties involved. Goals are high-level objectives of the business, stakeholders, or system. They are often used to discover, select, evaluate, and justify requirements for a service. Whereas functional requirements define the functions of the service, non-functional requirements (NFRs) characterize service properties and qualities, such as expected performance, reliability, availability, usability, and cost. Goals and NFRs capture essential (and often conflicting) system concerns and have a huge impact throughout the development process. In comparison with traditional software engineering, managing NFRs is even more difficult in a service engineering context due to the highly distributed nature of SOA applications where services are used in different business domains and under different contexts by a variety of stakeholders with little control over the services.

This chapter presents the User Requirements Notation (URN) as a means to model and analyze functional and non-functional service requirements. URN, a language standardized by the International Telecommunications Union (ITU-T), combines modeling concepts and notations for goals and intentions (mainly for non-functional requirements, quality attributes, and reasoning about alternatives) and scenarios (mainly for operational requirements, functional requirements, and performance and architectural reasoning) (Amyot, 2003; ITU-T, 2003; ITU-T, 2008a). In this chapter, we will tackle the modeling and analysis of NFRs for SOA from three perspectives, which can be used standalone or preferably combined:

a. The use of standard URN for modeling service goals and functions in an integrated way, hence enabling the analysis of NFRs in context. Current tool-supported techniques for scenario execution, strategy evaluation, and performance analysis will be briefly summarized.

b. The use of aspect-oriented extensions to URN (AoURN) to better modularize in individual units called concerns, different NFRs that can crosscut services or service components.

c. The use of new NFR-oriented annotations for service composition. These scenario annotations, based on a taxonomy also introduced in this chapter, enable the specification of quantitative NFR values for services and the computation of the NFR