Preface

Dynamic, trustworthy, and reliable service delivery in Service Oriented Architectures (SOA) is at present one of the main preconditions for successful and sustainable business operations. Service and business process reliability is, therefore, of paramount importance and cannot be compromised. Even today, services are simply expected to be delivered reliably and on demand, and this requirement will be even more important in the near future. Unreliable and incorrect services can corrupt business processes causing an impact ranging from lost opportunity or money to loss of lives. Our current understanding of service reliability properties is rather sketchy, limited, and mostly empirical. Regardless of this fact, services are already widely used as the new paradigm to build software architectures and enable complex business processes.

Several methodologies can be used to engineer reliable SOA systems: quantitative, qualitative, and analytical. Quantitative engineering is based on real-time measurement and monitoring. Whereas it has proven itself in several areas (e.g., hardware benchmarks and testing), it is difficult to apply to SOA because of the lack of adequate metrics and instrumentation. Qualitative engineering is performed informally (e.g., through questionnaires and interviews) and using best-practice frameworks (e.g., ITIL). Qualitative results are easy to misinterpret, difficult to compare, and depend heavily on the party that is performing the analysis. Analytical methods are used to model services and their behavior and calculate or simulate their reliability. Up to now, however, classical analytical methods have been applied to engineer service availability with mixed success and relatively low industry penetration due to scalability, complexity and evolution problems.

With the recent advance of cloud computing and software as a service paradigms, importance of the ability to model and assess service reliability has only increased. For example, with the technical (hardware and software) infrastructure growing much more complex in recent years, many SaaS providers have experienced an increasing number of outages, some of them also extremely long lasting. Consequently, fears were raised that inability to provide strictly defined service level agreement in terms of the maximum number of downtime hours per year may lead to rejection of the entire SaaS paradigm by the users. Many CTOs even speculated that they consider returning to locally hosted solutions, where SaaS products may be used as a backup application only. The increased usage of virtualization and server consolidation makes the problem even more acute, because virtual machine and/or application migration influence reliability and service level agreements of affected services, applications and processes.

Similar problems are evident in large scale SOA infrastructures (such as telecommunications sector or power grid communication). Recent blackouts and frequent failures of cellular communication services are just some of the numerous examples. Even worse, repair teams often depend on the same infrastructure, thus prolonging the failures.
These frequent problems vividly illustrate our inability to engineer complex and reliable SOA systems satisfactorily. They further indicate that reliable SOA systems must consider not only hardware and software properties, but also the underlying (network) infrastructure, configuration, and the people maintaining and using it.

This book attempts to address these issues and present a guide to the engineering of reliable SOA systems. It comprises three sections, treating service level agreements as the basic building block of reliable SOA systems, service composition as the mean of conquering complexity of SOA infrastructures with respect to reliability properties and finally several methods for design, and implementation of fault-tolerant and reliable SOA applications.

Proper specification of service level agreement (SLA) is the main pre-condition for successful implementation of non-functional properties, such as reliability. In Chapter 1, Massarelli et al. discuss the value of Service Level Agreements (SLA) used as contracts, and the authors propose an adaptive method for SLA runtime management that can monitor and perform changes during service execution phase. Di Modica and Tomarchio report in Chapter 2 that, although Service Level Agreements (SLAs) are the most common mechanism used to establish agreements on the quality of service, they have not been taken up by business stakeholders yet due to the low flexibility and usability, together with the lack of interoperability. The authors then propose an SLA management framework addressing these issues. In Chapter 3, Wang et al. present a service level management framework offering two benefits: incorporating monitoring and diagnosis services as an integral part of the QoS management. Zisman argues in Chapter 4 that creation, maintenance, and evolution of Service Level Agreements (SLAs) represent the main challenges in order to establish an effective evaluation and enforcement of quality of services and their providers. The author then presents an approach for SLA creation that is based on an iterative service discovery process. In Chapter 5, Hofer et al. present the work on configurability of non-functional properties in embedded systems. The authors propose a framework for making non-functional properties such as reliability, performance, or memory usage indirectly configurable and maintainable. Finally, Giallonardo and Zimeo explore in Chapter 6 the role that ontologies have on specifying service requirements, discovery and composition.

Service composition is the most straightforward method of managing complexity in SOA. However, service composition, with respect to non-functional properties, presents a new set of problems and challenges. In Chapter 7, Harney and Doshi explore Web Service composition in volatile environments, where parameters and properties of component services may change during execution. The authors propose a hierarchical Web Service composition mechanism as a solution. Blanco, Cardinale, and Vidal discuss in Chapter 8 the problem of identifying a service composition that best meets functional and non-functional user requirements. The authors propose a hybrid cost-based approach that receives and considers at the same time, a functional user request expressed by a pair of input and output attribute sets, and a non-functional condition represented by a set of QoS parameter permissible values. Cheikh proposes in Chapter 9 a service composition model considering available services that have access control constraints. As the scalability of service composition frameworks is one of the most limiting factors, the author investigates complexity in detail. In Chapter 10, Liu and Georgalas develop a novel approach for specifying non-functional properties of telecom operation support systems applications in the NGOSS framework. In Chapter 11, Ezenwoye and Sadjadi propose a language-based approach for transparently adapting BPEL processes in order to improve reliability. This approach addresses reliability at the business process layer using a code generator, which weaves fault-tolerant code to the original code and an external proxy.
In the world where on-demand and trustworthy service delivery is one of the main preconditions for successful business, service and business process availability is of the paramount importance and cannot be compromised. For that reason service availability is coming into central focus of the IT operations and management research and practice. Taweel investigates in Chapter 12 predictability of non-functional properties such as capability, reliability, compatibility, performance, et ceter. in complex service compositions. In Chapter 13, Milanovic and Milic present a framework for modeling business process and service availability. The approach takes into account services, the underlying ICT-infrastructure, and people. Sadi, Myers, and Sanchez investigate in Chapter 14 the failure analysis of SOA systems. They propose the metric for scanning a system model and flagging those components that are more likely to fail. In Chapter 15, Gönczy and Varró propose a model-driven approach to efficiently design and deploy standards-compliant service configurations with non-functional parameters, with special focus on reliable messaging. Looker and Munro argue in Chapter 16 that dependability assessment is an important aspect of any software system which shows the degree of trust and quality of service that is delivered by a system. They discuss service dependability and present different techniques that may be used in dependability assessment of SOA systems.

The book enhances our current understanding of service reliability and at the same time discovers new application possibilities, concepts, and challenges. Therefore, it can be read by both practitioners and researchers wishing to explore state-of-the art results from the field of reliable SOA application engineering.

Nikola Milanovic
Model Labs - Berlin, Germany