The promise of Service Oriented Computing technologies is a world of cooperating services where application components are assembled with little effort into end-to-end services that are loosely coupled to create business processes and flexible Service Oriented Architecture (SOA)-applications that span organizations and computing platforms. As such, Service Oriented Computing improves productivity by enabling enterprises to develop and bring new products and services to the market more rapidly.

An important characteristic of the Service Oriented Computing paradigm is that it enables application developers to dynamically grow application portfolios by creating compound SOA-application solutions that inter-mix internally existing organizational software assets with external services that possibly reside in remote locations. Understanding the nature of service composition is a daunting task. So far, the composition of distributed services is still far from being fully achieved. No effective, easy-to-use, flexible technology support is provided to assist in coping with many of the intricacies of service composition. These include, for instance, configuration of end-to-end service compositions by delivering the expected functionality with guaranteed multi-dimensional Quality of Service (QoS) levels, i.e., at both the application and system or IT-level, the ability to guarantee acceptable QoS levels for a composed service even when the operational conditions and regulations of its constituent services may continuously change.

An important area of services research gathering momentum is QoS-aware service composition. Service composition requires discovering service providers and delivering composed services that satisfy not only functional but also non-functional requirements meeting QoS constraints. For example, knowing that a service adopts a Web services security standard such as one from the stack of WS-Security specifications is not enough information to enable successful composition with other services. The service developer needs to know if the service actually requires WS-Security, what kind of security tokens it is capable of processing, and which one it prefers. Moreover, the developer must determine if the service should communicate using signed messages. If so, s/he must determine what token type must be used for the digital signatures. Finally, the developer must decide when to encrypt the messages, which encryption algorithm to use, and how to exchange a shared key with the service. For example, a purchase order service may indicate that it only accepts username tokens that are based signed messages using X.509 certificate that is cryptographically endorsed by a third party. Such considerations require understanding and respecting the component service policies, performance levels, security requirements, end-to-end Service Level Agreement (SLA) stipulations, and so forth.

When a developer designs a new SOA-application solution or performs a refactoring of an existing business process, determining the dependability and reliability of the final end-to-end composition of operations and QoS levels that make up the process is of paramount importance. These non-functional
(or QoS) requirements may describe essential performance and dependability requirements and apply across different logical layers of the application, from business-related details to system infrastructure; i.e., they are cross-cutting and considered multi-dimensional. Without considering QoS aspects of the business process such as determining inventory levels, delivery constraints, the maximum throughput, or time required to complete the process or its availability, the process will not meet the expectations of an enterprise or its clients. Achieving this effectively is a challenging problem as the service-oriented software is fully decentralized, and typically no single organization is in control of all the services involved in an SOA-application solution. Without a disciplined approach to non-functional properties of Service-oriented Systems, it would be impossible to understand how the software that will eventually run in production meets acceptable QoS levels, let alone automate the composition of business processes and mission critical applications.

This book deals with non-functional properties of Service-oriented Systems, which is a notoriously complex topic in a very sound and intuitive manner. It addresses a multitude of research problems in this field by providing a mixture of theoretical and practical solutions. The book addresses compelling problems such as service selection on the basis of user specified measurable QoS properties, methods for ranking services and selecting services that are part of larger execution chains, techniques for capturing delays that may occur during service provision and confirming whether required time-related properties are met, quality prediction techniques that support a service provider to determine possible QoS levels that can be guaranteed to a client in a manner that meets SLA expectations, contract reconciliation and SLA negotiation techniques, managing multi-dimensional SLAs, and many more interesting research issues. The book chapters are organized in four logical parts: Perspectives on Non-Functional Properties, Service Selection, Service Contracts, and SLA Governance.

This book covers an impressive number of topics and presents a wealth of research ideas and techniques that will excite any researcher (or practitioner) wishing to understand QoS management for Service-oriented Systems. It is pleasant to see that diverse and complex topics relating to QoS management are explained in an eloquent manner and include extensive references to help the interested reader find out more information about these topics. All in all this is an impressive book and an invaluable source of knowledge for advanced students and researchers working in or wishing to know more about this exciting field.

I commend the editors and the authors of this book on the breadth and depth of their work and for producing a well thought out and eminently readable book on such a complicated topic. Enjoy!

Michael P. Papazoglou
European Institute in Service Science, The Netherlands, July 2011

Michael Papazoglou is Scientific Director of the European Research Institute in Service Science (ERISS) and of the EC’s Network of Excellence, S-Cube. He is also an honorary Professor at the University of Trento in Italy, and Professorial Fellow at the Universities Lyon (France), New South Wales (Australia) and Rey Juan Carlos, Madrid (Spain). He has acted as an Adviser to the EC in matters relating to the Internet of Services and as a reviewer of national research programs for numerous countries around the world. His research interests lie in the areas of service oriented computing, web services, large scale data sharing, business and manufacturing processes, and distributed computing systems, where he has published 22 books (including monographs and conference proceedings), and well over 200 journal and conference papers with an H-index factor of 37.