Foreword

*Service-Oriented Computing* (SOC) is the computing paradigm that utilizes software services as fundamental elements for developing and deploying distributed software applications. Services are self-describing, platform-agnostic computational elements that support rapid, low-cost composition of distributed applications. They perform functions, which can be anything from simple requests to complicated business processes. Services allow organizations to expose their core competencies programmatically via a self-describing interface based on open standards over the network using standard languages (XML-based) and protocols.

A *Service System* is a system that depends on distributed control, cooperation, cascade effects, orchestration, and other emergent behaviors as primary compositional mechanisms to achieve its purpose. A service system’s purpose, structure, and number of components are increasingly unbounded in their development, use, and evolution. Service systems support the development of Web-scale service-based applications that are characterised by unbounded numbers and combinations of software-intensive, geographically dispersed, and globally available services.

*Service-driven Applications* are qualitatively different from traditional large-scale applications. They are typically realized by creating alliances between service providers, each offering services to be used or syndicated within other external services. They usually comprise aggregations of services in the form of end-to-end processes that cross organizational boundaries and can deliver effective solutions to the daunting challenges of *Enterprise Integration*.

Service development is continuously in flux. The pace of development has accelerated greatly due to the advanced requirements of service systems and the increasing complexity of application architectures (e.g. heterogeneous, virtualized, and increasingly Cloud-based). The standardization of *Service-driven Architectural Approaches* in conjunction with the advent of Cloud technologies offers new possibilities for developing pervasive Service-driven applications due to their flexibility and on-demand nature.

As a result of the technological advances described above, we are gradually experiencing the rise of “smarter service systems,” which become more versatile, flexible, mobile, dependable (i.e. continuously available, recoverable, and robust), energy-efficient, self-configurable, self-healing, and self-optimizing by adapting to changing operational contexts and environments.

As enterprises, governments, and non-profit organizations re-conceptualize their operations in terms of the Service-driven approaches and architectures, they will be well positioned for continuous efficiency improvements as well as capability expansions. The “double win” opportunity of both improved productivity and quality of service is for the first time feasible within this emerging computing paradigm.
Before software services can achieve their full potential and we can reap their benefits, the next generation of academia and professionals must master the fundamentals of Service-oriented Computing, and the technologies and Service-driven methodologies that underpin it. I am especially aware of the need to build this area on strong practical foundations. This book provides just such a foundation for software engineers, application developers, and enterprise architects.

The book covers an enormous wealth of important topics and technologies that mirror the evolution of software services. It provides an exhaustive overview of the challenges and solutions of major achievements pertaining to software services and architectures. Each chapter is an authoritative piece of work that synthesizes pertinent literature and highlights important accomplishments and advances in its subject matter. The content of this book is a testimonial of the leading role of its editors, authors, and reviewers and their highly influential work in the area of software services and architectures.

Chapter 1 overviews Service-driven architectural approaches which are a major driver for enabling agile, flexible, and extensible software applications and integration solutions that support today’s complex, multi-faceted and fast paced enterprises. It also focuses on architectural practices for implementing enterprise APIs into the overall Service-driven strategy of the enterprise.

Chapter 2 explores Enterprise Integration and examines its transition towards the current software service practices, and contemplates its evolution. Chapter 3 overviews Enterprise Integration approaches and focuses in particular on evaluating service architectures (e.g. Service-oriented Architecture (SOA) and Event-driven Architecture) in terms of complexity versus business benefit.

Chapter 4 deals with the important subject of processes and service compositions and proposes methods for resolving different kinds of message-level heterogeneity. Chapter 5 presents a methodological approach that improves the quality and cost-effectiveness of process-oriented, service-based applications by suggesting techniques on how to enrich business process models. Chapter 6 presents a framework to provide autonomous handling of long running business process transactions based on dependencies derived from the workflow. This chapter presents a solution for forward recovery from errors by applying compensation techniques to workflow enactments.

On the topic of processes and service compositions, Chapter 7 presents a model-driven design-time approach for access control in applications that comprise composed services. Chapter 8 explores the architectural practices and approaches such as service redundancy and design diversity, scaling, clustering, distributed data caching, in-memory data grid, and asynchronous messaging, for improving the dependability of services.

Chapter 9 overviews the topic of SOA Governance, and discusses tools, technologies, and practices employed to govern the service lifecycle in a Service-driven environment. Chapter 10 surveys the critical interdependencies between enterprise services and enterprise data and proposes best practices and strategies for coordinating the enterprise SOA and Enterprise Data Management approaches for mutual benefit. Chapter 11 provides practical guidance on business flexibility advantages obtained by careful management of vendor diversification options for enterprises that are implementing service-based applications and integration solutions.

Chapter 12 presents the important topic of Enterprise Mobile Service Architecture (EMSA), discusses the opportunities, and addresses mobile constraints and challenges in EMSA, which include aspects relating to mobile hardware, software, networking and mobility. Finally, Chapter 13 explores the capabilities and service models offered by the Cloud and the challenges of extending the Service-driven approach to the Cloud paradigm. This chapter presents design principles and implementation guidelines to architect application services in a Cloud ecosystem.
It is pleasant to see that diverse and complex topics relating to software services and architectures are explained in an eloquent manner and include important references to help the interested reader find out more information about these interesting topics.

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. All in all, this is an inspiring book and an invaluable source of knowledge for researchers and practitioners working on or wishing to know more about the exciting field of software services and enterprise architectures. It is well thought out and eminently readable!

Michael P. Papazoglou

European Research Institute in Service Science, The Netherlands

Michael P. Papazoglou is a full Professor and the Director of the European Research Institute in Service Science and the Scientific Director of the European Network of Excellence in Software Services and Systems (S-Cube) at Tilburg University, The Netherlands. He also holds honorary professorial positions in Australia, France, Italy, Spain, and Cyprus. His research interests include Service Oriented Computing, Cloud Computing, business processes, and federated and distributed information systems. He is one of the most cited researchers and internationally recognized for his research contributions in Service-oriented Computing. He has published 22 books and well over 200 journal and conference papers.