Preface

W3C defines a Web service as a software system designed to use standard protocols to support interoperable machine-to-machine interactions (publication, discovery, access, and orchestration) over a network. As Web services technology has matured in recent years, a new scalable Service-Oriented Architecture (SOA) is emerging as the basis for distributed computing and large networks of collaborating applications. To keep with the burgeoning field of web services and information technology, emerging and future research directions like the ones chronicled in this volume are the vital pulse that will keep researchers abreast of the latest and cutting edge.

Innovations, Standards and Practices of Web Services: Emerging Research Topics, edited by Liang-Jie Zhang, offers 17 contributions from authors from around the world, serving as experts in the vast and ever-expanding field of web services. This book has a thematic focus beyond the scope of the technology and language of computer mediated communication and interoperability, reaching instead to the topics of best practices, standards, and new ideas.

In Chapter 1, “Complex Network Theory Based Web Services Composition Benchmark Toolkit,” Seog-Chan Oh and Dongwon Lee present a novel benchmark toolkit, WSBen, that is capable of generating synthetic web services data with diverse scenarios and configurations using complex network theory. Web services researchers therefore can evaluate their web services discovery and composition algorithms in a more systematic fashion. The development of WSBen is inspired by their preliminary study on real-world web services crawled from the Web.

Chapter 2 is titled “USDL: A Service-Semantics Description Language for Automatic Service Discovery and Composition,” and it was written by Srividya Kona, Ajay Bansal, Luke Simon, Ajay Mally, Gopal Gupta, and Thomas D. Hite. The authors present an infrastructure using USDL (Universal Service-Semantics Description Language), a language for formally describing the semantics of Web services. USDL is based on the Web Ontology Language (OWL) and employs WordNet as a common basis for understanding the meaning of services. USDL can be regarded as formal service documentation that will allow sophisticated conceptual modeling and searching of available Web services, automated service composition, and other forms of automated service integration.

Chapter 3, “Privacy-Preserving Trust Establishment with Web Service Enhancements,” by Zhengping Wu and Alfred C. Weaver, proposes a mechanism whereby the service requester discovers the service provider’s requirements from a web service policy document, then formulates a trust primitive by associating a set of attributes in a pre-packaged credential with a semantic name, signed with the requester’s digital signature, to negotiate a trust relationship. Thus the service requester’s privacy is preserved because only those attributes required to build a trust relationship are revealed.
In Chapter 4, “Web Services Discovery with Rough Sets,” by Maozhen Li, Bin Yu, Vijay Sahota, and Man Qi, the authors present ROSSE, a Rough Sets based Search Engine for Web service discovery. One salient feature of ROSSE lies in its capability to deal with uncertainty of service properties when matching services. A use case is presented to demonstrate the use of ROSSE for discovery of car services. ROSSE is evaluated in terms of its accuracy and efficiency in service discovery.

Chapter 5, “A Model-Based Approach for Diagnosing Fault in Web Service Processes,” by Yuhong Yan, Philippe Dague, Yannick Pencolé, and Marie-Odile Cordier proposes a model-based approach to diagnose the faults in a Web service-composed business process. The authors convert a Web service orchestration language, more specifically BPEL4WS, into synchronized automata, to produce a formal description of the topology and variable dependency of the business process. After an exception is thrown, the diagnoser can calculate the business process execution trajectory based on the formal model and the observed evolution of the business process. The faulty Web services are deduced from the variable dependency on the execution trajectory.

Chapter 6, “Autonomous Web Services Migration in Mobile and Wireless Environments,” by Yeon-Seok Kim, Myung-Woo Park, and Kyong-Ho Lee, proposes an efficient method for migrating and replicating Web services among mobile devices. Specifically, the proposed method splits the source code of a Web service into subcodes depending on users’ preferences for its constituent operations. For the seamless provisioning of services, a subcode with a higher preference is migrated earlier than others. To evaluate the performance of the proposed method, the effect of the code splitting on migration was analyzed.

Chapter 7, “Estimating the Privacy Protection Capability of a Web Service Provider,” by George O.M. Yee, suggests that users would benefit from being able to choose (assuming that such estimates were made public) the service that has the greatest ability to protect their privacy (this would in turn encourage Web service providers to pay more attention to privacy). Web service providers would benefit by being able to adjust their provisions for protecting privacy until certain target capability levels of privacy protection are reached. This article presents an approach for estimating the privacy protection capability of a Web service provider and illustrates the approach with an example.

Chapter 8, “Issues on the Compatibility of Web Service Contracts,” by Surya Nepal and John Zic, categorizes compatibility issues in Web service contracts into two broad categories: (a) between contracts of different services (defined as a composability problem), and (b) a service contract and its implementation (defined as a conformance problem). This chapter examines and addresses these problems, first by identifying and specifying contract compatibility conditions, and second, through the use of compatibility checking tools that enable application developers to perform checks at design time.

Chapter 9, “High Performance Approach for Server Side SOAP Processing,” by Lei Li, Chunlei Niu, Ningjiang Chen, Jun Wei, and Tao Huang, proposes a new approach to improve Web services performance. Focusing on avoiding traditional XML parsing and Java reflection at runtime, this article presents a service-specific simple object access protocol (SOAP) processor to accelerate the execution of Web services. Moreover, the SOAP processor embeds several cache implementations and uses a novel adaptive cache mechanism, which can choose an optimized cache implementation dynamically. Through the experiments in this article, it is to be observed that the proposed approach can achieve a significant performance gain by incorporating the SOAP processor into the SOAP engine.

In Chapter 10, “A Framework and Protocols for Service Contract Agreements Based on International Contract Law,” by Michael Parkin, Dean Kuo, and John Brooke, a framework and a domain-independent negotiation protocol for creating legally binding contracts for service usage in a distributed, asynchronous service-oriented architecture is presented. The negotiation protocol, which builds on a simple agree-
ment protocol to form a multiround “symmetric” negotiation protocol, is based on an internationally recognized contract law convention. By basing our protocol on this convention and taking into account the limitations of an asynchronous messaging environment, we can form contracts between autonomous services across national and juridical boundaries, necessary in a loosely coupled, widely geographically distributed environment such as the Grid.

Chapter 11, “XML Data Binding for C++ Using Metadata,” by Szabolcs Payrits, Péter Dornbach, and István Zólyomi presents a novel way to map XML data to the C++ programming language. The proposed solution offers more flexibility and more compact code that makes it ideal for embedded environments. The article describes the concept and the architecture of the solution and compares it with existing solutions. This article is an extended version of the paper from ICWS 2006. The authors include a broader comparison with existing tools on Symbian and Linux platforms and evaluate the code size and performance.

Chapter 12, “The Assurance Point Model for Consistency and Recovery in Service Composition,” by Susan D. Urban, Le Gao, Rajiv Shrestha, Yang Xiao, Zev Friedman, and Jonathan Rodriguez, displays how their research has defined an abstract execution model for establishing user-defined correctness and recovery in a service composition environment. The service composition model defines a hierarchical service composition structure, where a service is composed of atomic and/or composite groups. The model provides multi-level protection against service execution failure by using compensation and contingency at different composition granularity levels. The model is enhanced with the concept of assurance points (APS) and integration rules, where APS serve as logical and physical checkpoints for user-defined consistency checking, invoking integration rules that check pre and post conditions at different points in the execution process.

Chapter 13, “Early Capacity Testing of an Enterprise Service Bus,” was written by Ken Ueno and Michiaki Tatsubori. This article proposes a capacity planning methodology and performance evaluation techniques for ESBs, to be used in the early stages of the system development life cycle. The authors actually run the ESB on a real machine while providing a pseudo-environment around it. In order to simplify setting up the environment we provide ultra-light service requestors and service providers for the ESB under test. They show that the proposed mock environment can be set up with practical hardware resources available at the time of hardware resource assessment. The experimental results showed that the testing results with the mock environment correspond well with the results in the real environment.

Chapter 14, “An Integrated Framework for Web Services Orchestration,” by C. Boutrous Saab, D. Coulibaly, S. Haddad, T. Melliti, P. Moreaux, and S. Rampacek, focuses on two features of Web services. The first one concerns the interaction problem: given the interaction protocol of a Web service described in BPEL, how to generate the appropriate client? Their approach is based on a formal semantics for BPEL via process algebra and yields an algorithm which decides whether such a client exists and synthesizes the description of this client as a (timed) automaton. The second one concerns the design process of a service. They propose a method which proceeds by two successive refinements: first the service is described via UML, then refined in a BPEL model and finally enlarged with JAVA code using JCSWL, a new language that we introduce here. Their solutions are integrated in a service development framework that will be presented in a synthetic way.

Chapter 15, “Security for Web Services: Standards and Research Issues,” by Lorenzo D. Martino, and Elisa Bertino, discusses the main security requirements for Web services and it describes how such security requirements are addressed by standards for Web services security recently developed or under development by various standardizations bodies. Standards are reviewed according to a conceptual
framework that groups them by the main functionalities they provide. Covered standards include most of the standards encompassed by the original Web Service Security roadmap proposed by Microsoft and IBM in 2002 (Microsoft and IBM 2002). They range from the ones geared toward message and conversation security and reliability to those developed for providing interoperable Single Sign On and Identity Management functions in federated organizations. The latter include Security Assertion Markup Language (SAML), WS-Policy, XACML, that is related to access control and has been recently extended with a profile for Web services access control; XKMS and WS-Trust; WS-Federation, Liberty Alliance and Shibboleth, that address the important problem of identity management in federated organizations. The chapter also discusses the issues related to the use of the standards and open research issues in the area of access control for Web services and innovative digital identity management techniques are outlined.

Chapter 16, “Web Service Enabled Online Laboratory,” by Yuhong Yan, Yong Liang, Abhijeet Roy, and Xinge Du, benchmarks the performance of the system when SOAP is used as the wire format for communication and propose solutions to optimize performance. In order to avoid any installation at the client side, the authors develop Web 2.0 based techniques to display the virtual instrument panel and real-time signals with just a standard Web browser. The technique developed in this paper can be widely used for different real laboratories, such as microelectronics, chemical engineering, polymer crystallization, structural engineering, and signal processing.

Chapter 17, “An Efficient Service Discovery Method and its Application,” by Shuiguang Deng, Zhaohui Wu, and Jian Wu proposes an information model for registered services. Based on the model, it brings forward a two-phase semantic-based service discovery method which supports both the operation matchmaking and operation-composition matchmaking. The authors import the bipartite graph matching to improve the efficiency of matchmaking. An implementation of the proposed method is presented. A series of experiments show that the method gains better performance on both discovery recall rate and precision than a traditional matchmaker and it also scales well with the number of services being accessed.

These 17 chapters present the latest research, perspectives, trends, and emerging issues in the field of web service research. They combine to become an essential reference for practitioners, academics, and students alike, serving as both a reference book and research manual.

Liang-Jie Zhang
Kingdee International Software Group Company Limited, China