Preface

The emergence and growth of Web technologies and applications has resulted in the need for specific methods for their integration. This collection, entitled *Web Services Research for Emerging Applications: Discoveries and Trends*, provides a comprehensive assessment of the latest developments in Web services research. Individual chapters focus on composing and coordinating Web services, XML security, and service oriented architecture, presenting new and emerging research in the Web services discipline.

Chapter 1, "*SOA Reference Architecture*," by Liang-Jie Zhang and Jia Zhang first analyzes the architectural style and metamodel of SOA Solution Stack (S3), then describes the various view models, and then describes an end-to-end SOA solution design and modeling framework and methodology.

Chapter 2, "*WSMoD: A Methodology for QoS-Based Web Services Design*," by M. Comerio, F. De Paoli, S. Grega, A. Maurino, and C. Batini presents WSMoD (Web Services MOdeling Design), a methodology that explicitly addresses this issue. Furthermore, it exploits general knowledge available on services, expressed by ontologies describing services, their qualities, and the context of use, to help the designer in expressing service requirements in terms of design artifacts.

Chapter 3, "*A Metamorphic Testing Methodology for Online SOA Application Testing*," by W. K. Chan, S. C. Cheung, and Karl R. P. H. Leung studies an approach to online testing. Service testing is divided into two steps. In the spirit of metamorphic testing, the offline step determines a set of successful test cases to construct their corresponding follow-up test cases for the online step. These test cases will be executed by metamorphic services that encapsulate the services as well as applicable metamorphic relations. Thus, any failure revealed by the approach will be a failure of the service under test.

Chapter 4, "*Integrated Design of eBanking Architecture*," by Tony C. Shan and Winnie W. Hua describes a service-oriented framework for integrated design of eBanking architecture (IDEA) in the financial services industry.

Chapter 5, "*A Similarity Measure for Process Mining in Service Oriented Architecture*," by Joonsoo Bae, Ling Liu, James Caverlee, Liang-Jie Zhang, and Hyerim Bae presents a quantitative approach to modeling and capturing the similarity and dissimilarity between different process designs. The authors derive the similarity measures by analyzing the process dependency graphs of the participating workflow processes. They first convert each process dependency graph into a normalized process matrix. Then they calculate the metric space distance between the normalized matrices. This distance measure can be used as a quantitative and qualitative tool in process mining, process merging, and process clustering, and ultimately it can reduce or minimize the costs involved in design, analysis, and evolution of workflow systems.

Chapter 6, "*Rapid Development of Adaptable Situation-Aware Service-Based Systems*," by S. S. Yau, S. Mukhopadhyay, H. Davulcu, D. Huang, R. Bharadwaj, and K. Shenai presents an approach to rapid development of adaptable situation-aware service-based systems. This approach is based on the α-logic and α-calculus, and a declarative model for situation awareness (SAW). This approach consists of four major components: (1) analyzing SAW requirements using the authors' declarative model for SAW, (2) translating the model representation to α-logic specifications and specifying a control flow graph in α-logic as the goal for situation-aware service composition., (3) automated synthesis of α-calculus terms that define situation-aware workflow agents for situation-aware service composition, and (4) compilation of α-calculus terms to executable components on an agent platform. An example of applying a framework in developing a distributed control system for intelligently and reliably managing a power grid is given.

Chapter 7, "*Object-Oriented Architecture for Web Services Eventing*," by Krzysztof Ostrowski, Ken Birman, and Danny Dolev proposes an architecture based on object-oriented design principles that is free of such limitations, extremely modular and extensible, and that can serve as a basis for extending and complementing the existing standards. The new approach emerges from the authors' work on Live Distributed Objects, a new programming model that brings object-orientation into the realm of distributed computing.

Chapter 8, "*Composing and Coordinating Transactional Web Services*," by Frederic Montagut, Refik Molva, and Silvan Tecumseh Golega proposes a new process to automate the design of transactional composite Web services. This solution for Web services composition does not take into account functional requirements only but also transactional ones based on the Acceptable Termination States model. The resulting composite Web service is compliant with the consistency requirements expressed by business application designers and its execution can easily be coordinated using the coordination rules provided as an outcome of the authors' approach. An implementation of these theoretical results augmenting an OWL-S matchmaker is further detailed as a proof of concept.

Chapter 9, "*Security Personalization for Internet and Web Services*," by George O.M. Yee and Larry Korba derives the content of an Internet or Web service security policy and proposes a flexible security personalization approach that will allow an Internet or Web service provider and customer to negotiate to an agreed-upon personalized security policy. In addition, this chapter presents two application examples of security policy personalization, and an overview of the design of the authors' security personalization prototype.

Chapter 10, "*XML Security with Binary XML for Mobile Web Services*," by Jaakko Kangasharju, Tancred Lindholm, and Sasu Tarkoma considers ways to integrate an alternate format with security, and presents a solution that the authors see as a viable alternative. In addition to this, the chapter presents extensive performance measurements, including ones on a mobile phone on the effect of an alternate format when using XML-based security. These measurements indicate that, in the wireless world, reducing message sizes is the most pressing concern, and that processing efficiency gains of an alternate format are a much smaller concern. The authors also make specific recommendations on security usage based on their measurements.

Chapter 11, "*Efficient and Effective XML Encoding*," by Christian Werner, Carsten Buschmann, Ylva Brandt, and Stefan Fischer discusses the most relevant state-of-the-art technologies for compressing XML data. Furthermore, it presents a novel solution for compacting SOAP messages. In order to achieve significantly better compression rates than current approaches, the compressor described in this chapter utilizes structure information from an XML Schema or WSDL document. With this additional knowledge on the "grammar" of the exchanged messages, this compressor generates a single custom pushdown automaton, which can be used as a highly efficient validating parser as well as a highly effective compressor. The main idea is to tag the transitions of the automaton with short binary identifiers that are then used to encode the path through the automaton during parsing. The authors' approach leads to extremely compact data representations and is also usable in environments with very limited CPU and memory resources.

Chapter 12, "*Data Mining in Web Services Discovery and Monitoring*," by Richi Nayak recommends several data mining applications that can leverage problems concerned with the discovery and monitoring of Web services. This chapter then presents a case study on applying the clustering data mining technique to the Web service usage data to improve the Web service discovery process. This chapter also discusses the challenges that arise when applying data mining to Web services usage data and abstract information.

Chapter 13, "*A Reengineering Approach for Ensuring Transactional Reliability of Composite Services*," by Sami Bhiri, Walid Gaaloul, and Claude Godart presents a reengineering approach to ensure transactional reliability of composite services. Contrary to previous approaches which check correctness properties based on the composition model, the authors start from executions' log to improve services' recovery mechanisms. Basically, the chapter proposes a set of mining techniques to discover the transactional behavior from an event based log. Then, based on this mining step, the authors use a set of rules in order to improve services' reliability.

Chapter 14, "*Karma2: Provenance Management for Data-Driven Workflows*," by Yogesh L. Simmhan, Beth Plale, and Dennis Gannon discusses the growing need for data-driven e-Science applications that are under the control of workflows composed of services on the Grid. The focus of this work is on provenance collection for these workflows that are necessary to validate the workflow and to determine quality of generated data products. The challenge the authors address is to record uniform and usable provenance metadata that meets the domain needs while minimizing the modification burden on the service authors and the performance overhead on the workflow engine and the services. The framework is based on generating discrete provenance activities during the lifecycle of a workflow execution that can be aggregated to form complex data and process provenance graphs that can span across workflows. The implementation uses a loosely coupled publish-subscribe architecture for propagating these activities, and the capabilities of the system satisfy the needs of detailed provenance collection. A performance evaluation of a prototype finds a minimal performance overhead (in the range of 1% for an eight-service workflow using 271 data products).

Chapter 15, "*Result Refinement in Web Services Retrieval Based on Multiple Instances Learning*," by Yanzhen Zou, Lu Zhang, Yan Li, Bing Xie, and Hong Mei presents a new approach to improve this kind of category-based Web Services retrieval process which can refine the coarse matching results step by step. The refinement is based on the idea that operation specification is very important to service reuse. Therefore, a Web Service is thus investigated via multiple instances view in this approach, which indicates that a service is labeled as positive if and only if at least one operation provided by this service is usable to the user. Otherwise, it is labeled as negative. Experimental results demonstrate that this approach can increase the retrieval precision to a certain extent after one or two rounds of refinement.

Chapter 16, "*A Model-Driven Development Framework for Non-Functional Aspects in Service Oriented Architecture*," by Hiroshi Wada, Junichi Suzuki, and Katsuya Oba proposes a model-driven development (MDD) framework for non-functional aspects in SOA. The proposed MDD framework consists of (1) a Unified Modeling Language (UML) profile to graphically model non-functional aspects in SOA, and (2) an MDD tool that accepts a UML model defined with the proposed profile and transforms it to application code. This chapter also demonstrates how the proposed framework is used in model-driven development of service-oriented applications. Empirical evaluation results show that the proposed MDD framework improves the reusability and maintainability of service-oriented applications by hiding low-level implementation technologies in UML models.

Chapter 17, "*Interoperability Among Heterogeneous Services: The Case of Integration of P2P Services with Web Services*," by Aphrodite Tsalgatidou, George Athanasopoulos, and Michael Pantazoglou addresses the interoperability problem by first presenting its multiple dimensions and then by describing a conceptual model called Generic Service Model (GeSMO), which can be used as a basis for the development of languages, tools and mechanisms that support interoperability. The authors illustrate how GeSMO has been utilized for the provision of a Peer-to-Peer (P2P) service description language and a P2P invocation mechanism which leverages interoperability between heterogeneous P2P services and between P2P services and Web services.

Chapter 18, "*Service-Oriented Architecture for Migrating Legacy Home Appliances to Home Network System: Principle and Applications*," by Masahide Nakamura, Hiroshi Igaki, Akihiro Tanaka, Haruaki Tamada, and Ken-ichi Matsumoto presents a practical framework that adapts the conventional home electric appliances with the infrared remote controls (legacy appliances) to the emerging home network system (HNS). The proposed method extensively uses the concept of service-oriented architecture to improve programmatic interoperability among multi-vendor appliances. The authors first prepare APIs that assist a PC to send infrared signals to the appliances. Then the APIs are aggregated within self-contained service components, so that each of the components achieves a logical feature independent of device/vendor-specific operations. The service components are finally exhibited to the HNS as Web services. As a result, the legacy appliances can be used as distributed components with open interfaces. To demonstrate the effectiveness, the authors implement an actual HNS and integrated services with multi-vendor legacy appliances. The authors also show practical applications implemented on the developed HNS.

Chapter 19, "*Broadening JAIN-SLEE with a Service Description Language and Asynchronous Web Services*," by Paolo Falcarin, Claudio Venezia, and José Felipe Mejia Bernal describes the StarSLEE platform which extends JAIN-SLEE in order to compose JAIN-SLEE services with Web services and the StarSCE service creation environment which allows exporting value added services as communication Web services. It analyzes open issues that must be addressed to introduce Web Services in new telecom service platforms.

Chapter 20, "*Workflow Discovery: Requirements from E-Science and a Graph-Based Solution*," by Antoon Goderis, Peter Li, and Carole Goble presents the case for workflows and workflow discovery in science and develop one discovery solution. Through a survey with 21 scientists and developers from the myGrid/Taverna workflow environment, workflow discovery requirements are elicited. Through a user experiment with 13 scientists, an attempt is made to build a benchmark for workflow ranking. Through the design and implementation of a workflow discovery tool, a mechanism for ranking workflow fragments is provided based on graph sub-isomorphism detection. The tool evaluation, drawing on a corpus of 89 public workflows and the results of the user experiment, finds that, for a simple showcase, the average human ranking can largely be reproduced.

Chapter 21, "*An Access Control Framework for WS-BPEL Processes*," by Federica Paci, Elisa Bertino, and Jason Crampton intoduces a new type of WS-BPEL activity to model human activities and by developing RBAC-WS-BPEL, a role based access control model for WS-BPEL and BPCL, a language to specify authorization constraints.

Chapter 22, "*Business Process Control-Flow Complexity: Metric, Evaluation, and Validation*", by Jorge Cardoso focuses on investigating process complexity. A metric to analyze the control-flow complexity of business processes is presented and described. The metric is evaluated in terms of Weyuker's properties in order to guarantee that it qualifies as good and comprehensive. To test the validity of the metric, this chapter describes the experiment that was carried out for empirically validating the metric.

Chapter 23, "*Pattern-Based Translation of BPMN Process Models to BPEL Web Services*," by Chun Ouyang, Marlon Dumas, Arthur H.M. ter Hofstede, and Wil M.P. van der Aalst advances the state of the art in BPMN-to-BPEL translation by defining methods for identifying not only perfectly block-structured fragments in BPMN models, but quasi-structured fragments that can be turned into perfectly structured ones and flow-based acyclic fragments that can be mapped onto a combination of structured constructs and control links. Beyond its direct relevance in the context of BPMN and BPEL, this chapter addresses issues that arise generally when translating between graph-oriented and block-structured flow definition languages.

Chapter 24, "*DsCWeaver: Synchronization-Constraint Aspect Extension to Procedural Process Specification Languages*," by Qinyi Wu, Calton Pu, Akhil Sahai, and Roger Barga presents DSCWeaver, a tool that enables a synchronization-aspect extension to procedural languages. It uses DSCL (directed-acyclic-graph synchronization constraint language) to achieve three desirable properties for synchronization modeling: fine granularity, declarative syntax, and validation support. DSCWeaver then automatically generates executable code for synchronization. The authors demonstrate the advantages of their approach in a service deployment process written in BPEL and evaluate its performance using two metrics: lines of code (LoC) and places to visit (PtV). Evaluation results show that this approach can effectively reduce the development effort of process programmers while providing performance competitive to unwoven BPEL code.

Chapter 25, "*A Reservation-Based Extended Transaction Protocol for Coordination of Web Services within Business Activities*," by Wenbing Zhao, Firat Kart, L. E. Moser, and P. M. Melliar-Smith describes a novel reservation-based extended transaction protocol for coordination of tasks within a business activity. With the advance of Web Services, the authors anticipate that business activities will be implemented as Web Services and that the automation of business activities across multiple enterprises over the Internet will become a reality. Classical extended transaction protocols are not well suited for this new breed of business activity, because of their use of compensation transactions. The authors' reservation-based extended transaction protocol employs an explicit reservation phase and an explicit confirmation/cancellation phase to eliminate the need for compensation transactions. The authors show how their Reservation Protocol maps to the Web Services Coordination specification and describe their implementation of the Reservation Protocol.

Web Services Research for Emerging Applications: Discoveries and Trends aims to provide contemporary coverage of Web services with a focus on how these systems help to define the future of computing and communication. Through investigations of different applications, processes, and protocols, this collection aims to inform researchers of new and emerging trends in the development and research of Web services.