Preface

Future software development in a global environment, based on service provision, discovery, and composition, will be substantially influenced by two important areas: Service Oriented Architecture (SOA)—with a strong background in research communities—and Cloud Computing—with its Software as a Service component and mainly promoted by industrial players.

Generally, scientific literature that presents principles, methods, and tools for engineering service-oriented systems assumes that these systems are developed from scratch, even if this is rarely the state of practice. Because of economic and social constraints, preserving legacy systems is a necessity. Moreover, the adoption of newly developed systems for service provisioning may be costly in terms of money, time, and human resources. Therefore, experience has proved that a gradual transition is more efficient, and a well-conducted migration can have beneficial effects on the evolution of an organization.

Service Oriented Architecture (SOA) enables a system to respond easily to new requirements, and to assimilate new business services and new service providers while the business is developing. Services may be created for processing data, streamlining and reusing functionality incorporated in legacy systems, and integrating activities performed by multiple businesses or government partners. The SOA architecture supports a wide distribution of the deployed software artifacts. Agility and extensibility are increased with the use of services discovered at design time, or even runtime with the use of semantic technologies. SOA is meant to reduce system maintenance efforts and to prevent current and future integration problems via loose coupling between service providers and consumers implied by the composition of reusable and replaceable services.

Cloud Computing is emerging as a new computational model in which software is hosted, run, and managed in large server farms and data centers and provided as a service over the Web. Users of cloud services are exonerated from software licensing, installation, and maintenance, which would be necessary if the software was executed on their own computers. The dream of providing computing as a utility has been made easier by two emerging technologies: virtualization and Software as a Service (SaaS). Depending on the content of the service, a cloud can also offer Infrastructure as a Service (IaaS)—raw computing services such as CPU and storage, and Platform as a Service (PaaS)—COTS products, tools, and middleware for developing and deploying applications.

Migration to SOA and cloud environments can be challenging. From a SOA perspective, the challenges come from the nature of the legacy systems being migrated, as well as the characteristics of the SOA environment targeted for migration. Legacy system challenges include poor separation of concerns, tool availability, architectural mismatch, operational mismatch, and dependencies on commercial products. SOA environment characteristics can also create challenges for migration, including size of the service consumer community, internal vs. external service consumers, open vs. closed SOA infrastructure tech-
nologies, and legacy system operations. From a cloud perspective, the challenges include technology mismatch, calculating cost/benefit analysis, security, data privacy, and legal issues. However, the challenges in both cases have to be evaluated against the opportunities of system modernization, potential for serving a larger number of users, scalability, and systems interoperability and integration.

Chapter 1 of this book, “Introduction to the Migration from Legacy Applications to Service Provisioning,” analyzes the global context that led to the major change in the ownership paradigm—the shift towards provisioning and consuming services—and to the need to make changes, yet preserve legacy, so to migrate applications gradually towards service orientation. The chapter offers a background for understanding the solutions, the research challenges, and the experiments presented in this book, pointing out some landmarks of migration to SOA environments, specifics of cloud environments, and an opening towards other approaches based on services. Hints for navigating through the book chapters are also provided.

After that, the book is organized in three more sections: Section 2, “Migrating to SOA Environments,” Section 3, “Migrating to Cloud Environments,” and Section 3, “Migrating to Service-Oriented Systems: Frontier Approaches.”

MIGRATING TO SOA ENVIRONMENTS

This section contains literature reviews, research roadmaps, technical solutions, strategies, and practical experiences regarding the migration of legacy applications to SOA environments.

Chapter 2, “Research Challenges in the Maintenance and Evolution of Service-Oriented Systems,” presents the research agenda of the domain, including current approaches, as well as challenges, gaps, and needs for future work. Five important topics are identified and analyzed: (1) tools, techniques, and environments to support maintenance activities; (2) multilanguage system analysis and maintenance; (3) reengineering processes for migration to SOA environments; (4) transition patterns for service-oriented systems; and (5) runtime monitoring of service-oriented systems. Chapter 3, “Legacy to SOA Evolution: A Systematic Literature Review” offers a historical overview of migration of business-critical legacy applications to SOA systems, based on platform- and language-independent interfaces and abstraction of underlying logic. The chapter describes the review protocol, including research question, data sources, search strategy, study selection strategy, data extraction, and data synthesis. It also presents the evaluation framework, followed by a discussion on findings, best practices, and open research issues. Chapter 4, “Reengineering and Wrapping Legacy Modules for Reuse as Web Services: Motivation, Method, Tools, and Case Studies,” starts with an analysis of business and technical reasons for reusing legacy code and migrating to SOA environments. It focuses on transforming legacy modules into Web services, introducing selection criteria, and transformation methods and tools, followed by outlining the lessons learned in three modernization projects. Chapter 5, “Service Identification and Specification with SoaML,” maps service design elements onto SoaML and describes the necessary steps for creating services: (1) requirements analysis – including data models, use cases, and business processes; (2) service identification – for fulfilling the necessary functionalities and aligning with business and with legacy systems; and (3) specification of service interfaces, messages types, and data types. Chapter 6, “The SOA Frontier: Experiences with Three Migration Approaches,” deals with the set of registered Web Services that represent the frontier between consumer and provider. It presents three migration approaches that were applied for modernizing a government agency system written in COBOL: (1) wrapping for
developing Web services; (2) redesign and reimplementation; and (3) an optimized combination. The assessment of cost and time required for performing refactoring and the discussion on service interface quality represent important contributions of this chapter. Chapter 7, “Model-Driven Software Migration: Process Model, Tool Support, and Application,” analyzes the state-of-the-art regarding migration methods and selects one derived from the horseshoe model and enriched by extensively applying model-driven engineering. It introduces a four-phase process composed of preparation, conceptualization, migration, and transition, and presents a customizable tool suite supporting the migration.

**MIGRATING TO CLOUD ENVIRONMENTS**

This section presents strategies, solutions, and experiments for migrating existing applications and data to cloud environments, taking full advantage of the virtualization, scalability, and elasticity potential.

Chapter 8, “Moving to SaaS: Building a Migration Strategy from Concept to Deployment,” analyzes the parallel between SOA as an architectural model, and Software as a Service as a delivery or business model. It questions the efficiency of applying approaches for migration to SOA to migration to SaaS, outlining shortcomings related to quality of service, security, configurability, scalability, and multi-tenancy. The chapter proposes a global migration strategy, from business to technology, with a first phase dedicated to evaluation and decision-making, and a second phase for implementation and service provisioning, ending with the presentation of a practical use case. Chapter 9, “Migration of Data between Cloud and Non-Cloud Datastores,” considers the need to migrate legacy data, which accompanies application migration and also involves a transformation of the underlying schemas. It proposes an approach based on an RDF/RDFS intermediate model, with detailed mapping algorithms and their proofs, plus performance evaluation. Chapter 10, “Migrating a Legacy Web-Based Document-Analysis Application to Hadoop and HBase: An Experience Report,” considers a lexical-analysis tool implemented in Ruby with SOAP4R Libraries as the source of migration, and two potential target solutions for data storage. The solutions are compared with respect to the development effort (e.g. lines of code and conceptual difficulty) and the method of performance (e.g. time required to construct indices and number of operations per minute achieved for representative services). It discusses criteria for selecting between solutions based on this comparison. Chapter 11, “Geographically-Distributed Cloud-Based Collaborative Applications,” describes a triple transformation, first to a logical server as a cluster of virtualized servers, then to a multi-server application in cloud, and finally to a multi-cloud application with geographically distributed clusters in order to improve end-to-end latency.

**MIGRATING TO SERVICE-ORIENTED SYSTEMS: FRONTIER APPROACHES**

This section looks at the vision of service-orientation beyond SOA and Cloud, providing an insight on their connections with REST architectures and Service-Oriented Computing in general.

Chapter 12, “Bridging the SOA and REST Architectural Styles,” introduces a hierarchy of architectural styles, and presents a structural service style that combines the constraints and the complementary characteristics inherited from Service-Oriented Architecture and Representational State Transfer (REST) styles. Chapter 13, “Considerations of Adapting Service-Offering Components to RESTful Architectures,” considers the case of migrating legacy SOA applications to REST architectures, and defines a
full adaptation framework accompanied by a process model. Chapter 14, “Model-Driven Integration of Heterogeneous Software Artifacts in Service–Oriented Computing,” introduces a parallel between integrating legacy artifacts in Cloud Computing environments and integrating devices in Ubiquitous Computing environments using service-orientation principles such as separation between provider and consumer parts. Because the migration of a legacy application may represent a complex endeavor, there is also the possibility to reuse existent software artifacts and integrate them in a redefined service-oriented system. The approach aims to manage these reused legacy artifacts, proposing a platform that would allow their integration regardless of their original type, equally supporting iPOJO, WS-Spring, and UPnP, and having the possibility of extending to any service-oriented, component-based and service-oriented component-based systems.

TARGET AUDIENCE

The book assumes that its readers have a background in software engineering and basic knowledge of Service-Oriented Computing.

Areas of Interest

The goal of this book was to provide an opportunity for closer collaboration between the SOA and Cloud Computing communities and for identifying potential solutions to the challenges raised by current business environment dynamics. In addition, the goal was to offer an insight into related research originating from other approaches concerned with the migration of legacy data and applications to new, emerging technologies for comparing the experience gained in creating services, architectures, models, and methods.

Types of Professionals

Service-oriented systems bring together a large variety of stakeholders, including technical and managerial staff; in addition, educational programs at all levels are concerned with this topic in response to a growing occupational sector. This book targets professionals involved in developing, maintaining, or studying software based on services, including researchers, practitioners, as well as PhD students and university professors whose research areas pertain to services science.

Technical Levels

The main purpose of the book is to grasp the most advanced principles and techniques of migration to service provisioning by presenting state-of-the-art strategies, roadmaps, methods, tools, and real-life and academic experiments. However, it is also important to cover all technical levels, from introductory to expert, so that each reader can navigate through the book to discover the large number and scale of issues related to reusing and migrating legacy systems and obtain answers appropriate for his or her particular expertise.
CONCLUSION

The endeavor of editing this book revealed that research on migrating legacy applications to SOA and Cloud environments is spread all over the world, both in academic and industrial organizations. This topic is often blended with reengineering and maintenance, but it emerges as a distinct, up-to-date concern, which has its own agenda, and whose future growth can be definitely foreseen. We hope this book will awake or meet your already existing interest in the achievements, best practices, and challenges of modernizing existing information systems by converting them to a service-provisioning paradigm.

Anca Daniela Ionita
University “Politehnica” of Bucharest, Romania

Marin Litoiu
York University, Canada

Grace Lewis
Carnegie Mellon Software Engineering Institute, USA