At Your Service

Service-Oriented Computing from an EU Perspective

edited by Elisabetta Di Nitto, Anne-Marie Sassen, Paolo Traverso, and Arian Zwegers

The MIT Press
Cambridge, Massachusetts
London, England

For information about special quantity discounts, please email special_sales@mitpress.mit.edu

This book was set in Times New Roman and Syntax on 3B2 by Asco Typesetters, Hong Kong.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

At your service : service-oriented computing from an EU perspective / edited by Elisabetta Di Nitto . . . [et al].
p. cm. — (Cooperative information systems)
Presents results of collaborative research projects of the European Community's Information Society Technologies Programme about service-oriented computing.
Includes bibliographical references and index.
ISBN 978-0-262-04253-6 (hardcover : alk. paper) 1. Web services—Research—Europe. 2. Internetworking (Telecommunication) 3. Application program interfaces (Computer software) 4. Business enterprises—Computer networks. I. Di Nitto, Elisabetta. II. Information Society Technologies Programme.
TK5105.88813.A85 2009
006.7′6—dc22 2008027498

10 9 8 7 6 5 4 3 2 1

1 Introduction

Elisabetta Di Nitto, Anne-Marie Sassen, Paolo Traverso, and Arian Zwegers

1.1 Context

Service-oriented computing (SOC) represents one of the most challenging promises to support the development of adaptive distributed systems. Applications can open themselves to the use of services offered by third parties that can be accessed through standard, well-defined interfaces. The binding between applications and the corresponding services can be extremely loose in this context, thus making it possible to compose new services on the fly whenever a new need arises.

Around this idea a number of initiatives and standards have grown up, some of them focusing on how such roles need to interact with each other, and some others on how to engineer systems based on such a model and on how to provide foundational support to their development. In particular, so-called service-oriented applications and architectures (SOAs) have captured the interest of industry, which is trying to exploit them as the reference architecture to support B2B interaction. In this context, according to Forrester Research, the SOA service and market had grown by $U.S. 4.9 billion in 2005, and it is forecasted to have an interesting growth rate until 2010, with a compound annual growth rate of 88 percent between 2004 and 2009.[1]

In this scenario, with the actual adoption of SOA currently being at the level of focused experiments, a number of issues and challenges need to be addressed over both the long and the short term. Some of them are the following:

- The need to support the governance of complex SOA-based systems
- The need to properly address physical and conceptual interoperability among different integrated systems
- The need to properly guarantee the dependability of a system composed of parts owned by third parties
- The need to build approaches that support the (self-) adaptiveness of applications built upon the SOA model
- The opportunity to exploit distributed computational capabilities to achieve high-level objectives

- The possibility to adopt the SOA approach in contexts different from the classical B2B one and more oriented to the development of pervasive, multidevice systems.

In all these areas the research community is experimenting with different solutions and approaches, and we are witnessing the evolution of a melting pot of ideas and intuitions, coming from different fields, into more established and well-structured approaches. This is why we were asked to provide an overview on what is happening in the research projects funded by the European Community in this area, within the Information Society Technologies (IST) Sixth Framework Programme (FP6).[2]

The objectives of IST in FP6 are to ensure European leadership in domain-independent as well as domain-specific technologies at the heart of the knowledge economy. IST aims to increase innovation and competitiveness in European businesses and industries, and to contribute to greater benefits for all European citizens. Thus, the focus is on the future generation of technologies in which computers and networks will be integrated into the everyday environment, rendering accessible a multitude of services and applications through easy-to-use human interfaces. This vision of ambient intelligence places the user, the individual, at the center of future developments for an inclusive knowledge-based society for all.

Service-oriented computing is one of the research fields IST is focusing on, since it embodies relevant challenges for the vision of ambient intelligence, and appears to be one of the most promising topics for both research and industrial exploitation. For the projects that have been involved in the writing of this book, IST has contributed 94 million euros out of a total cost of these projects of about 153 million euros during 2002–2006. Such an investment, together with the establishment of the NESSI technology platform on services,[3] shows the willingness of Europe to act as a main player both at the institutional and at the industrial level.

This book shows how some of the most significant efforts funded by the European Community in the period when it was being written are addressing the open issues mentioned above. The presentation is structured to cover different aspects related to the development and operation of a service-based system. In the remainder of this chapter we briefly review the terminology that is used in the rest of the book, we provide an overview of the layers typical of a service-oriented application, and, finally, we outline the structure of the book.

1.2 Main Definitions and Terminology

As the various chapters of this book will show, services and SOAs can be used for purposes ranging from the most common applications on B2B integration to pervasive and autonomic systems. Of course, this means that the main concepts and definitions can vary from case to case. We have been trying to keep a uniform terminology by selecting the conceptual model for services (Colombo et al. 2005) developed within the SeCSE project (see chapter 10) as a common reference, and we have discovered that it should be extended

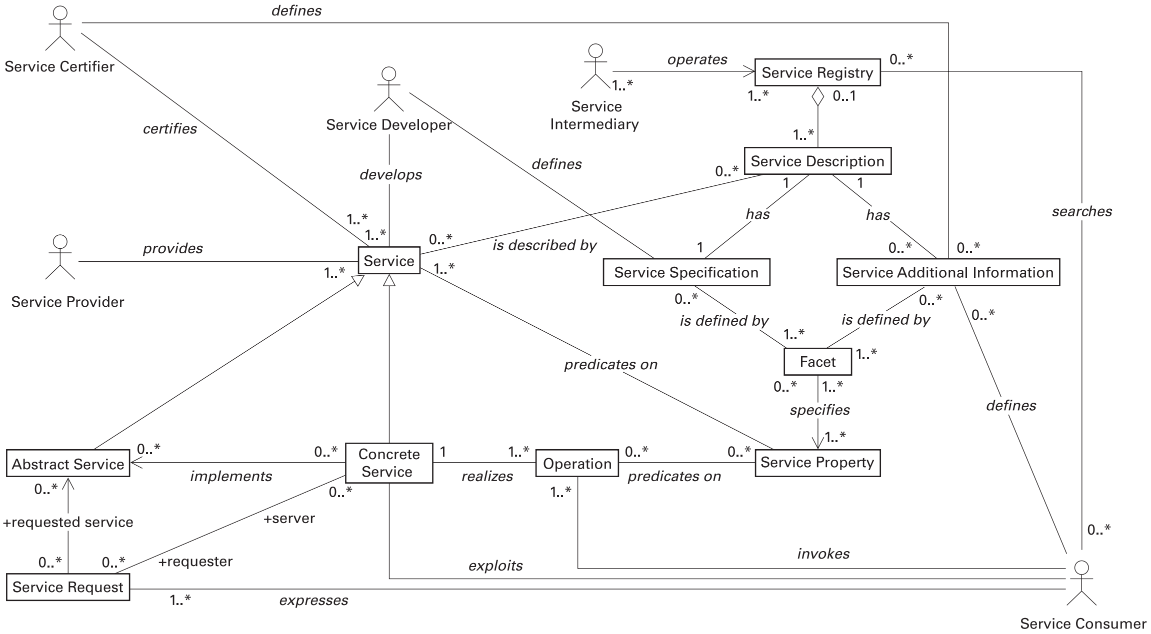

Figure 1.1
The core diagram of the SeCSE conceptual model

with new concepts in various directions to make it fully adequate to cover all concepts introduced in the various contributions. Its main core, however (see figure 1.1), remains unchanged, and represents the characteristic concepts of a service-oriented application. *Services*, in the context of this book, always represent the production of some result upon request. Such production is often performed by a software system that implements the service. However, it could also be performed by a human being or by cooperation between humans and computers.

In general, a service has a description associated with it. In the case of SeCSE such a description is organized in two parts, one that is defined when the service is designed and that contains all information needed to operate properly with the service (the service specification), and another that can be optionally used to incorporate additional information about the service. The service specification can include the signature of service operations, as well as behavioral constraints and properties, nonfunctional characteristics and so-called SLA templates that define the structure of the service-level agreements that the service can fulfill. The additional information contains information about the execution of services and is usually fed by the service users or by a third party, for instance, the party in charge of certifying the service. In SeCSE, service specifications and additional information are organized in facets, each of which provides a piece of the whole information about the service.

As will be discussed mainly in chapter 4, services can be implemented by using technologies ranging, for instance, from Web Services to P2P and Grid Services. Also, they can have specific characteristics, such as being dependable (see chapter 19) or collaborative (see chapter 5), and therefore can be exploited by multiple users at the same time through multiple interfaces and asynchronous and peer-to-peer interaction modes. Semantic services have an associated ontological description, provided in proper languages such as OWL-S or WSMO (see chapters 12, 13, and 14). Thanks to this description, the activities that require discovering and assembling compositions of services are greatly simplified. The explicit definition of semantics also allows the designer to focus more easily on a specific application domain, as happens in chapter 14, which focuses on services for the smart home application domain. Finally, services can be conversational (see chapters 8 and 9), as when they are used for business-to-business integration. In this context, interoperability issues have to be addressed as well (see chapter 18).

Software systems implementing the services can be autonomic or self-adaptive, in the sense that they are able to sense and adapt themselves, and their ability to offer services under different environmental conditions (see chapters 2, 6, and 9). Discovery of services also can happen by taking into account the environment and the context in which services operate (see chapters 3 and 4). Finally, services can be created not only by developers but also by end users (see chapter 11), by providing tools that allow for an easy end-user interaction.

1.3 Layers of a Service-Oriented Platform

Service-oriented computing is a research field that includes multiple aspects and facets. We have classified the chapters in this book according to two different and independent taxonomies. The first one is developed by the NESSI platform.[4] It aims to provide a unified view of European research in services architectures and software infrastructures. This view will be the basis for defining technologies, strategies, and deployment policies fostering new and open industrial solutions and societal applications that enhance the safety, security, and well-being of citizens.

The second taxonomy is "the SOC road map" (Papazoglou et al. 2006), which has been developed by researchers in the service-oriented community. It aims at identifying the specific research challenges arising mainly in the area of service integration. It is therefore more specific than the first one, and will be used to refine the classification of the projects dealing with service integration. The rest of this section describes the two taxonomies in more detail.

1.3.1 The NESSI Framework

Figure 1.2 shows the taxonomic framework of NESSI. It stretches from low-level infrastructure considerations (e.g., dealing with large-scale resource virtualization) to high-level

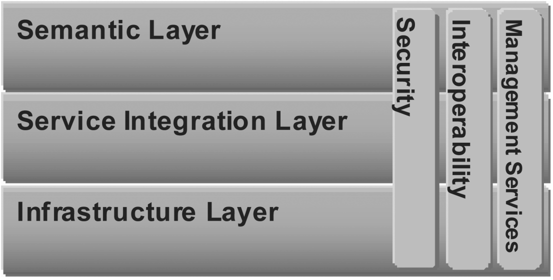

Figure 1.2
The NESSI framework

semantic considerations (e.g., supporting the definition and use of ontologies, reasoning with situations, and representing intentions). The framework also addresses concerns that span all these different levels of abstraction, covering cross-cutting issues such as security, system and service management, and interoperability. Below, we provide some details on the layers that are part of the road map. The reader should note that NESSI is using "service" in a broad sense, including network connectivity, computing hardware, application environments, and data and information, as well as application components.

Infrastructure The infrastructure domain aims at the virtualization of resources across servers, storage, distributed systems, and the network. Infrastructures have to be designed and implemented to be robust, fault-tolerant, and secure. From a user's perspective, infrastructures must be transparent (almost invisible) during the entire life cycle—allowing a plug-and-play approach to infrastructure use as well as to Grid provisioning and operation of services.

The internals of the infrastructure layer should be, in principle, completely transparent to the services being executed on top of it. The upper layer can, however, gain advantage from the Grid resource replication mechanisms and its autonomic capabilities for load balancing, clustering of similar/diverse components, and so on.

Service Integration In the NESSI vision, SOAs will become the primary architecture for business systems of the near future. SOAs provide means to create complex systems in a new modular way, simply by configuration and composition. This modularity will allow reusability of published services by other applications within a virtual organization paradigm. In this context, the general problem of configuring and composing a set of services, at both the functional and the business levels, is a difficult one: dependable systems can be built only from reliable configurations and compositions. Therefore, the service integration layer also aims at providing tools and methods for configuration and composition in the same way that existing CASE tools provide support for programming.

Moreover, the layer should also support dynamic reconfiguration by allowing software to be modified without stopping execution. The potential is great—dynamic reconfiguration allows systems to evolve and extend without loss of service, thus meeting the demands for high availability.

Semantics Semantics is a key element for the transformation of information into knowledge. One way to build knowledge will be through advanced search engines that allow fast search in a large body of unstructured data. Semantic Web technology based on ontologies will enable far more effective machine-to-machine communication.

On the business process level, business modeling provides the semantics that is required for business process management, process transformation, and intercompany cooperation. In a knowledge-based economy, learning and knowledge management finally will have to converge to a workplace utility.

Security and Trust Concern over security is one of the most significant barriers to acceptance of IT and digital services as a utility, becoming even more crucial in a highly dynamic environment. Security and trust in a utility-driven world can be achieved only by an end-to-end perspective that addresses all layers involved. An example is the consistent treatment of identity (of people, resources, and processes) balanced with mechanisms for providing levels of privacy and anonymity where required by the legal or regulatory environment or by user wishes. Related to this is the need for a practical yet rigorous approach to trust in large distributed systems, as well as models and mechanisms for secure and trusted interenterprise cooperation and cooperation in virtual organizations.

Management Services Central to the NESSI vision of a service-oriented utility are automated and autonomic management techniques for efficient and effective management of large, dynamic systems. These will include the following:

- Service life cycle management to support identification of components, their location, negotiation, and reservation, as well as their orchestration, configuration, and operational management. Life cycle management will also have to support the withdrawal and release of resources, accounting, and settlement.
- Trust, service-level agreements (SLAs), or contract management that deals with aspects such as the agreement over Quality of Service (QoS) provisioning mapped to SLAs, flexible QoS metrics, and the management of QoS violation. In addition, common principles for defining unambiguous SLAs that can be associated with a measurement and audit methodology will be necessary in a commercial environment.
- Managing of the complexity (including emergent properties) of global-scale, distributed ICT so that performance can be predicted and controlled.
- Mechanisms for controlled sharing of management information, end-to-end coordination, and performance prediction and management.

Interoperability and Open Standards Interoperability, including the use of open standards, when understood in its widest sense involves any kind of ICT at any level. It deals with aspects such as interfaces between different systems, abstraction between layers, connectivity, standardized protocols, interoperability to support dynamic composition of services, business process interfaces, standards for interenterprise cooperation, and integration with sensors and other devices. Industry-developed open standards will constitute the key mechanisms to overcome the current interoperability problems that generate frustration and distrust in new technology.

1.3.2 The SOC Road Map

The SOC road map (shown in figure 1.3) separates basic service capabilities provided by a services middleware infrastructure and conventional SOA from more advanced service functionalities that are needed for dynamically composing (integrating) services. It also

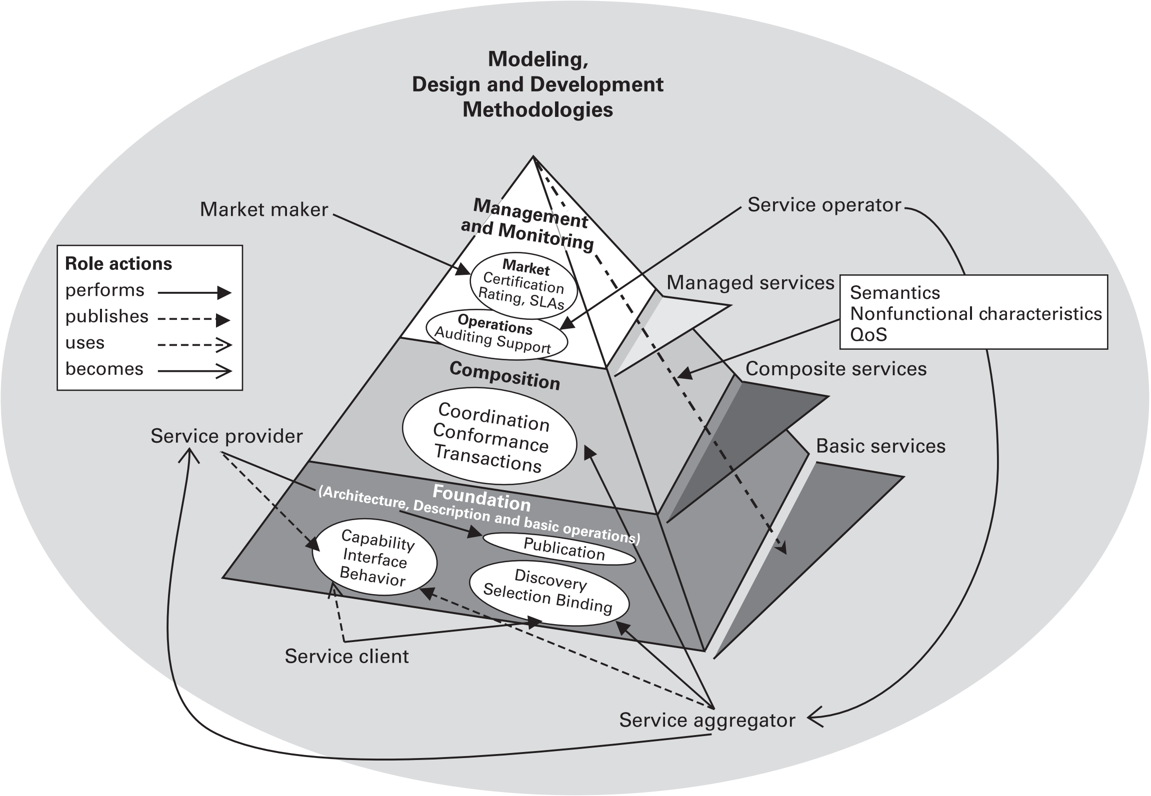

Figure 1.3
The SOC roadmap

distinguishes between a layer for composing services and a layer for the management of services and of their underlying infrastructure.

As shown in the figure, there are three planes. The bottom plane utilizes the basic service middleware and architectural constructs for describing, publishing, and discovering services, and the service composition and management planes are layered on top of it. The perpendicular axis indicates service characteristics that cut across all three planes. These include semantics, nonfunctional service properties, and quality of service (QoS). Quality of service encompasses important functional and nonfunctional service quality attributes, such as performance metrics (response time, for instance), security attributes, (transactional) integrity, reliability, scalability, and availability. Delivering QoS on the Internet is a critical and significant challenge because of its dynamic and unpredictable nature.

Figure 1.3 also highlights the importance of service modeling and service-oriented engineering (i.e., service-oriented analysis, design, and development techniques and methodologies that are crucial elements for the development of meaningful services and business process specifications). Service-oriented engineering activities help in developing meaningful services, service compositions, and techniques for managing services. Service engineering is thus applied to the three service planes shown in figure 1.3.

1.4 Structure of the Book

The chapters of this book have been organized according to the taxonomy shown in figure 1.4. The taxonomy was obtained by merging the two visions and road maps described in

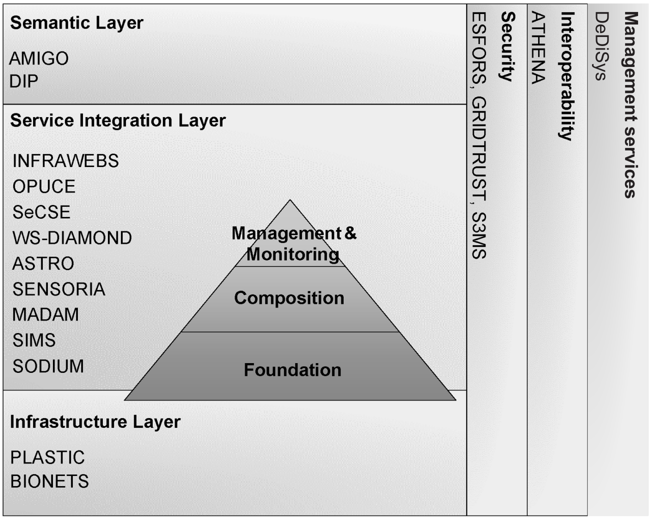

Figure 1.4
The structure of the book

the previous section. Indeed, the SOC road map can nicely detail one of the layers of the NESSI framework (i.e., the service integration layer).

The two projects that address the service infrastructure layer are BIONETS and PLASTIC (chapters 2 and 3). The emerging trends toward pervasive computing and communication environments will make it possible for user-oriented services to interface directly with the surrounding environment. This opens up enormous opportunities for context-related services, but also poses problems related to the management of these complex networks and services. Chapter 2 describes the biologically inspired approach of BIONETS on how to provide a fully integrated network and service environment that scales to large amounts of heterogeneous devices, and that is able to adapt and evolve in an autonomic way.

Pervasive computing will enable users to access information and computational resources anytime and anywhere. Such a setup is highly open and dynamic: pervasive computing systems should support ad hoc deployment and execution, integrating the available hardware and software resources at any given time and place. Service discovery will thus have an important role to play. Chapter 3 describes possible service discovery mechanisms in pervasive, heterogeneous environments.

Chapter 4 (on the SODIUM project) also deals with service discovery and heterogeneity, but at the service integration level of the NESSI framework. It provides innovative solutions for the description, discovery, and composition of heterogeneous services such as Web Services, Grid Services, or P2P services. In fact, SODIUM proposes an abstraction layer that hides the technical details of each service type from both developers and end users, without altering or modifying the distinct properties and characteristics of the underlying technologies.

Chapter 5, which describes the results of SIMS, introduces the concept of semantic interfaces for defining the collaborative behavior of services and the goals that can be achieved through collaborative behavior. Semantic interfaces are used to compose services and to ensure compatibility between the collaborating parts of the services at design time and runtime.

Chapter 6, which describes results of the MADAM project, proposes development approaches for self-adaptable, context-aware applications based on separation of concerns. Applications developed according to this approach dynamically optimize the end-user experience according to the context of the (mobile) user.

Chapter 7, which presents the results of the SENSORIA project, provides mathematically founded and sound methodologies and tools for dealing with the amount of flexibility and interoperability needed in next-generation service-oriented architectures. SENSORIA aims to support a more systematic and scientifically well-founded approach to engineering of software systems for service-oriented overlay computers.

Chapters 4, 5, 6, and 7 all concern the foundation layer of the SOC road map, since they provide basic principles and theory for describing and discovering services. Chapter 8, about the ASTRO project, covers a substantial part of the composition layer of the SOC road map, focusing essentially on the provision of innovative and powerful orchestration functionalities. ASTRO provides design-time support to composition, as well as runtime support by allowing the monitoring of composite services.

Chapter 9, on WS-DIAMOND, concerns the management layer of the SOC road map, since it describes a framework for self-healing Web Services.

The last category of project contributions to the service engineering issues identified by the SOC road map and the service integration layer of NESSI is represented by the SeCSE, OPUCE, and INFRAWEBS projects. All three projects provide useful principles and supporting tools to develop service-oriented systems. Chapter 10 (SeCSE) focuses on service description of both functional and nonfunctional properties of services, and how they can be composed into dependable service compositions. Chapter 11 (OPUCE) describes how end users can adapt and compose telecommunications services according to their needs while they are on the move. It tries to bring Web 2.0 principles to the telecommunications world. Chapter 12 (INFRAWEBS) describes a methodology and tools to develop semantic Web Services, either by starting from scratch or by converting nonsemantic Web Services into semantic Web Services.

Although placed in the semantics layer of the NESSI framework, chapter 13 is highly related to chapter 12, dealing also with the development of semantic Web Services. In particular, it describes some of the results of the DIP project and shows the real potential of semantics in a compelling real-world application in the e-government domain. Chapter 14, about the AMIGO project, shows how semantics may be used to realize the ambient intelligence vision of a networked home.

Chapters 15–19 deal with cross-cutting issues. Chapters 15, 16, and 17 deal with security. Chapter 15 describes the ESFORS project, which has identified security requirements for service-oriented systems and has done a survey on relevant research to address the requirements. Chapter 16 describes a fine-grained access control proposed by the GridTrust project that addresses the security challenges created by the next-generation grids where computing and storage power and services are shared among a (dynamic) pool of users. In chapter 17 the notion of security by contract (S × C), developed by the S3MS project, is explained as a mechanism to create trust when nomadic users would like to download services that are useful for their context at a particular moment.

Chapter 18 presents the contribution of the ATHENA project within the cross-cutting issue of interoperability. In particular, ATHENA has developed an interoperability framework and a reference architecture that relate the areas in which interoperability issues can occur. They range from the business level down to the implementation level.

Chapter 19's focus is on dependability. If service-oriented computing is ever going to be used for mission-critical systems, the ability to prove the dependability of the system is crucial. The DeDiSys project proposes dependability techniques for both data-centric and resource-centric services.

Chapters 20 and 21 present an outlook to the future. How will all these valuable results be taken up by industry in order to create innovation and new jobs? In chapter 20, NESSI explains how this industry-driven initiative plans to take up these results and to fill the gaps.

Of course the issues that remain unsolved do not relate only to the industrial adoption of the results that have been developed so far. Basic research still needs to be further developed. In order to support this effort, in the first call of the new Framework Programme for Research (FP7) the European Commission has agreed to co-fund the S-Cube network of excellence in service-oriented computing. S-Cube's plans for further research are presented in chapter 21.

Acknowledgment

The work published in this chapter is partly funded by the European Commission under the Sixth Framework Programme. The work reflects only the authors' views. The Commission is not liable for any use that may be made of the information contained therein.

Notes

1. IDC, Enterprise Integration & SOA Conference, 2007. http://www.idc.com/italy/events/ei07/ei07.jsp.
2. http://cordis.europa.eu/ist/.
3. www.nessi-europe.eu.
4. http://www.nessi-europe.com/Nessi/Portals/0/Nessi-repository/Publications/Flyers/2006_02_NESSI_SRA_VOL_1.pdf.

References

M. Colombo, E. Di Nitto, M. Di Penta, D. Distante, and M. Zuccalà. 2005. Speaking a common language: A conceptual model for describing service-oriented systems. In *Service-Oriented Computing—ICSOC 2005: Third International Conference.... Proceedings*, ed. B. Benatallah, Fabio Casati, and Paolo Traverso. LNCS 3826, pp. 50–62. Berlin: Springer Verlag.

M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. 2006. Service-oriented Computing Research Roadmap. http://infolab.uvt.nl/pub/papazogloump-2006–96.pdf.