Grant Agreement N° 215483

Title:	*Creation of course registries*
Authors:	*Lero, UniDue, UoC, VUA, TUW, USTUTT, INRIA, SZTAKI, Tilburg, CITY, UPM, TUW*
Editor:	*Angela Kounkou (CITY)*
Reviewers:	*Olivier Barais (INRIA)*
	Mohand-Said Hacid (UCBL)
Identifier:	*CD-SoE-1.2.6*
Type:	*Deliverable*
Version:	*1*
Date:	*17 March 2010*
Status:	*External*
Class:	*Internal*

Management Summary

This document reports on the initial set of learning material collected as part of S-Cube Spreading of Excellence (SoE) activities. WP-SoE 1.2 ("Community Outreach") aims among others to disseminate S-Cube research outcomes among the wider scientific community, notably using electronic means to reach individuals and institutions external to the consortium.

This deliverable reports on learning material that was collected from S-Cube partners with a view to making it available on an online platform. It provides an overview and a description of the initial set of material currently available, and highlights gaps in the current collection that ought to be addressed by subsequent collection within S-Cube.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 215483 (S-Cube).

Members of the S-Cube consortium:

University of Duisburg-Essen (Coordinator)	Germany
Tilburg University	Netherlands
City University London	U.K.
Consiglio Nazionale delle Ricerche	Italy
Center for Scientific and Technological Research	Italy
The French National Institute for Research in Computer Science and Control	France
Lero - The Irish Software Engineering Research Centre	Ireland
Politecnico di Milano	Italy
MTA SZTAKI – Computer and Automation Research Institute	Hungary
Vienna University of Technology	Austria
Université Claude Bernard Lyon	France
University of Crete	Greece
Universidad Politécnica de Madrid	Spain
University of Stuttgart	Germany
University of Hamburg	Germany
Vrije Universiteit Amsterdam	Netherlands

Published S-Cube documents

All public S-Cube deliverables are available from the S-Cube Web Portal at the following URL:

http://www.s-cube-network.eu/results/deliverables/

The S-Cube Deliverable Series

Vision and Objectives of S-Cube

The Software Services and Systems Network (S-Cube) will establish a unified, multidisciplinary, vibrant research community which will enable Europe to lead the software-services revolution, helping shape the software-service based Internet which is the backbone of our future interactive society.

By integrating diverse research communities, S-Cube intends to achieve world-wide scientific excellence in a field that is critical for European competitiveness. S-Cube will accomplish its aims by meeting the following objectives:

- Re-aligning, re-shaping and integrating research agendas of key European players from diverse research areas and by synthesizing and integrating diversified knowledge, thereby establishing a long-lasting foundation for steering research and for achieving innovation at the highest level.
- Inaugurating a Europe-wide common program of education and training for researchers and industry thereby creating a common culture that will have a profound impact on the future of the field.
- Establishing a pro-active mobility plan to enable cross-fertilisation and thereby fostering the integration of research communities and the establishment of a common software services research culture.
- Establishing trust relationships with industry via European Technology Platforms (specifically NESSI) to achieve a catalytic effect in shaping European research, strengthening industrial competitiveness and addressing main societal challenges.
- Defining a broader research vision and perspective that will shape the software-service based Internet of the future and will accelerate economic growth and improve the living conditions of European citizens.

S-Cube will produce an integrated research community of international reputation and acclaim that will help define the future shape of the field of software services which is of critical for European competitiveness. S-Cube will provide service engineering methodologies which facilitate the development, deployment and adjustment of sophisticated hybrid service-based systems that cannot be addressed with today's limited software engineering approaches. S-Cube will further introduce an advanced training program for researchers and practitioners. Finally, S-Cube intends to bring strategic added value to European industry by using industry best-practice models and by implementing research results into pilot business cases and prototype systems.

S-Cube materials are available from URL: http://www.s-cube-network.eu/

Table of Contents

1 Introduction 5

1.1 Workpackage vision 5
1.2 S-Cube research topics 5

2 Collected learning material 6

2.1 Engineering 6
2.1.1 Advanced Internet Computing 6
2.1.2 Service Oriented Design 6
2.1.3 Introduction to Model-Driven Engineering 7
2.1.4 Web Services in the Semantic Web 7
2.1.5 Quality of Service for Web Services 7
2.1.6 A QoS Ontology and a Semantic QoS Metric Matching Algorithm 7
2.1.7 QoS-Based Web Service Discovery 8
2.1.8 Software Quality & Processes 8
2.1.9 Quality Assurance for Service-based Systems: From Software Engineering to Service Engineering 8
2.1.10 Human Provided Services in Mixed Service-Oriented Systems 9
2.1.11 Applying Social Network Analysis Techniques to Software Engineering 9
2.1.12 An Introduction to the Analysis of Computational Systems through Abstract Interpretation 9
2.1.13 Service-Centric Systems and Requirements Engineering 9
2.1.14 Service-based software development 10
2.1.15 A framework for proactive dynamic service discovery 10
2.2 Business process Management 10
2.2.1 Web Services Business Process Execution Language (WS-BPEL) 10
2.2.2 Agile Service Networks 10
2.3 Infrastructure and Middleware 11
2.3.1 Introduction to Grid computing 11

3 Summary and conclusion 11

1 Introduction

In this deliverable, we provide an overview of the learning modules collected from S-Cube partners and aimed for use in an online platform provided by S-Cube: the Virtual Campus. This section outlines the vision for the Spreading of Excellence activities as pertaining to this deliverable (section 1.1), and introduces the learning material collected from S-Cube partners and its initial classification (section 1.2).

The remainder of this document is organized as follows. Section 2 presents the method and templates used to collect learning material and describes each of the learning items; section 3 provides an overview of the collected material and concludes this deliverable; and the appendix[1] lists the collected, full-length learning materials.

1.1 Workpackage vision

As specified in the S-Cube Description of Work, spreading of excellence activities are scheduled as part of S-Cube's Joint Programme of Activities, and are split into two work packages that, among other, aim at the training of researchers and the dissemination of S-Cube research results. More specifically, WP-SoE-1.1 (Virtual Campus) aims to structure and develop training and education programmes on the topic of service-oriented computing while WP-SoE-1.2 (Community outreach) focuses on spreading S-Cube research among the wider scientific and industrial communities, notably using the S-Cube web portal for dissemination. This deliverable, although a part of WP-SoE-1.2, also provides inputs for SoE 1.1. by collecting and presenting learning material which will be used in Masters and PhD programmes made available on the Virtual Campus.

1.2 S-Cube research topics

S-Cube's Joint Research Activities are distributed between workpackages addressing different topics, some of which are cross-cutting across the JRAs. More specifically, the JRAs and their corresponding workpackages are as follow:

- **JRA-1: Engineering and Adaptation Methodologies for Service-based Systems**, which focuses on developing an integrated set of principles, techniques and methodologies for engineering, adapting and monitoring hybrid service-based applications, and further ensuring end-to-end quality provision and SLA conformance. This comprises the following workpackages:
  - WP-JRA-1.1 - engineering principles, techniques and methodologies for hybrid, service-based applications - which focuses on exploiting knowledge from fields other than service science for the engineering of SBAs
  - WP-JRA-1.2 - adaptation and monitoring principles, techniques and methodologies for service-based applications - which focuses on defining novel principles for the cross-layer monitoring of service-based systems.
  - WP-JRA-1.3 - end-to-end quality provision and SLA conformance - which focuses on defining principles, techniques and methodologies for end-to-end, cross-layer quality provision and SLA conformance
- **JRA-2: Realization Mechanisms for Service-based Systems**, which focuses on developing novel mechanisms to guarantee the seamless operation of functional SBA layers: business process management (BPM), service composition and coordination (SCC), and service infrastructure (SI). JRA-2 comprises the following workpackages:
  - WP-JRA-2.1 - business process management - which focuses on concepts for service implementation from business models and for better collaboration and decision-making within integrated Agile Service Networks.

[1] Presented as full-size slides to preserve the legibility of the resources which would be lost otherwise.

- WP-JRA-2.2 - Adaptable Coordinated Service Compositions - which focuses on service composition and coordination
- WP-JRA-2.3 - Self-* Service Infrastructure and Service Discovery Support - which focuses on policies and techniques for self-adaptive and self-healing services, and on registry support mechanisms.

Aside from their workpackage assignment, the topics covered in these JRAs can be classified according to the following broad topics of affinity:

- *Engineering*, including all phases of services and SBAs lifecycle (as presented in the S-Cube lifecycle model) and interweaving HCI and context factors throughout as and where relevant: requirements engineering, design, construction and quality assurance, deployment and provisioning, operation, management and quality assurance, and adaptation.
- *Business process Management*, which includes concepts and definitions of BPM-related concepts (e.g. business process, business activity, rules, transactions, protocols). It further specifies the modeling, execution, integration, analysis, monitoring and optimization of business processes.
- *Infrastructure and Middleware*, which detail the technical foundations upon which much of the research in other work packages is based (e.g. Grid computing, infrastructure and network concepts).

This first level of classification between the different courses is used in this deliverable to present the collected learning modules, establish their relation to the S-Cube research topics, and assist in the identification of little-represented research areas in this initial set. The classification will be refined as the module collection continues and learning programmes are structured from partners' contributions.

2 Collected learning material

Course material pertaining to Software Services and Systems and collected from S-Cube partners is presented in the following sub-sections. The items' content, aims, and pre-requisites (where appropriate) are summarised in the abstracts provided below; the full contents are available in the appendices. Unless otherwise specified, the target audience are Master students with a background in Computer Science, software, service or systems engineering, or related disciplines.

2.1 Engineering

2.1.1 Advanced Internet Computing

This course discusses theoretical foundations, technologies, architectures, standards and examples of recent developments regarding Internet Computing and their applications. The focus is on Service-oriented Computing and Web services; a good programming knowledge in Java is required and the following topics are covered:

- Service Oriented Computing and Service-oriented Architectures
- Enterprise Application Integration and Middleware
- Web services - Foundations, Architectures and Standards (SOAP, WSDL, UDDI)
- Web services Composition and Workflows (e.g., BPEL, WS-Coordination, WS-Transaction, BPML, WSCI etc.)

2.1.2 Service Oriented Design

This module presents advanced design techniques applicable to large service-oriented and software systems. The lectures present innovative software and service-oriented design techniques, and emphasis is given among others to service identification and SOA design and migration.

- **Lecture 1**: this lecture introduces services and their key aspects. Concepts of SOA, service discovery, service composition, QoS and SLAs are presented, and key differences between system development and service oriented development are detailed.
- **Lecture 2**: Service Oriented analysis - the process of determining how business requirements can be represented through services – is presented along with service modeling guidelines. Service Oriented design (the process of modeling a service inventory and/or reusing it to compose a SOA) is then detailed.
- **Lecture 3**: finally, this lecture introduces the use of views to illustrate important service aspects for service-oriented software systems; two industrial case studies are presented: HomeFutura, a digital home application and BancaFutura, a banking organization.

2.1.3 Introduction to Model-Driven Engineering

This set of 5 lectures introduces modeling as a way to master the complexity of modern software development and presents concepts and principles of model-driven engineering.

- **Lecture 1**: this lecture presents UML and Model Driven Engineering, and introduces Model Driven Architecture as a special case of Aspect Oriented Design.
- **Lecture 2**: this lecture discusses object-oriented meta-languages for model-driven engineering and presents the Kermeta language
- **Lecture 3**: lecture 3 introduces model transformation techniques and discusses model-to-text and model-to-model approaches
- **Lecture 4**: this lecture discusses software product lines (SPL) and the application of model driven engineering to SPL
- **Lecture 5**: finally, lecture 5 discusses aspect oriented model driven engineering.

2.1.4 Web Services in the Semantic Web

The Semantic Web provides technologies to support a machine-interpretable web, while Web Services and Service-Oriented Computing in general offer a standard way of offering capabilities via a medium such as the Internet. The Semantic Web Services vision aims to combine the Semantic Web and Web Services technologies in order to enable automatic and dynamic interaction between software systems. This lecture mainly deals with the issue of describing Semantic Web Services, introduces the concept of ontologies as a tool for the creation of enriched WS descriptions towards the achievement of semantic web services, and presents ontology technologies.

2.1.5 Quality of Service for Web Services

This set of 4 lectures, structured to be followed in series, presents concepts and principles relating to modeling quality of service in web services and service compositions. The QoS issues for web services are studied, focusing on the design of service compositions which have associated a Service Level Agreement based on quality characteristics of the services. The QoS models and metamodels are discussed, evaluation of quality of service for service compositions and design of service compositions with guaranteed quality properties are discussed.

- **Lecture 1**: Introduction to quality of service representation
- **Lecture 2**: Quality of service optimization in web services
- **Lecture 3**: Design of dependable service-based processes
- **Lecture 4**: Testing quality of service compositions
- **Lecture 5**: Quality of service negotiation

The course's target audience are Master students in software, service or systems engineering with a basic background in engineering processes and techniques.

2.1.6 A QoS Ontology and a Semantic QoS Metric Matching Algorithm

Various QoS ontologies have been proposed in the literature that can be used to describe the QoS capabilities of a Web Service and the QoS requirements of Web Service requesters. In this

presentation, one of the richest of such QoS ontologies - OWL-Q - is presented and its usage is highlighted by the proposal of a novel semantic QoS metric matching algorithm that can be used in QoS-based WS discovery algorithms in order to increase their precision and recall. The target audience are Master or PhD Students in software engineering, service engineering, and computer science; a basic knowledge of standard WS and Semantic Web technologies is required.

2.1.7 QoS-Based Web Service Discovery

Various QoS-based Web Service discovery algorithms have been proposed in the literature. However, they present specific drawbacks which prevent their wide use and adoption. In this presentation, novel algorithms for QoS-based Web Service matchmaking and selection are analyzed. Moreover, the empirical evaluation conducted on specific implementations of these algorithms, that use different constraint solving techniques, is presented. The target audience are Master or PhD Students, or young researchers in software engineering, service engineering, and computer science. A basic knowledge of the standard WS technologies is expected but not required.

2.1.8 Software Quality & Processes

This set of 5 lectures, structured to be followed in series, presents concepts and principles relating to software quality and processes. It is important for software engineers, regardless of the environment for which they are developing software, to be aware of the quality and process issues which can arise. Therefore, these lectures are dealing with software quality and processes in general, not focusing specifically on the development of software for services. However, the concepts and principles which they discuss should be considered when developing software for services.

- **Lecture 1**: This lecture discusses quality of any product, focusing specifically on a framework for Total Quality Management. We then examine how software fits into each of the 'fitness' defined within Total Quality Management.
- **Lecture 2**: In this lecture, we present why process quality is important, discussing the advantages and disadvantages of implementing and improving processes within the software organization.
- **Lecture 3**: In lecture 3, we discuss how to make a software process work – through the implementation of process thinking, process focus and process discipline. Following the presentation of each of these concepts, the management of an effective process is discussed.
- **Lecture 4**: Following on from Lecture 3, the material presented here looks at the implementation of the effective software process – what must be in place to ensure that the process results in software product quality.
- **Lecture 5**: Without measurement, it is very difficult to understand where our software process weaknesses are. This lecture discusses different kinds of measurement which can be used when measuring software process quality.

2.1.9 Quality Assurance for Service-based Systems: From Software Engineering to Service Engineering

To assure the desired quality of a service-based application, two complementary strategies can be employed: constructive and analytical quality assurance. Where the goal of constructive quality assurance is to prevent the introduction of faults (or defects) while the artefacts are created (in the sense of 'correctness by construction'), the goal of analytical quality assurance is to uncover faults in the artefacts after they have been created.

In this module we motivate the need for quality assurance of service-based systems and provide an introduction to the three major classes of approaches for analytical quality assurance in service-based applications: (i) **Testing**, the goal of which is to (systematically) execute services or service-based applications with predefined inputs in order to uncover failures, (ii) **Monitoring**, which observes services or service-based applications as well as their context during their current execution, (iii) **Static Analysis**, the aim of which is to systematically examine (without execution) an artefact (e.g., a

service specification) to determine certain properties or to ascertain that some predefined properties are met.

The module, in large parts, draws on the body of knowledge of software engineering and based on this knowledge highlights the key differences and key research challenges for quality assurance of service-based systems.

2.1.10 Human Provided Services in Mixed Service-Oriented Systems

While traditional SOA tend to only involve software services, today's systems are compositions of services and people communicating in an anytime-anywhere manner thanks to pervasive network access and evolutions in the Web (such as Web 2.0) enabling more people than ever to collaborate (notably in Web communities axed around common interests) and to create and consume electronic content. This lecture presents the challenges and interactions in mixed systems. It further describes the Human -Provided Services (HPS) framework, which supports interactions between humans, interactions with services, and enables humans to publish their skills and capabilities as Web services. Finally, this lecture presents trust challenges and mechanisms to address them in the context of mixed systems which participants flexibly join and leave, with highly dynamic interactions that are often influenced by the role and reputation of collaboration partners.

2.1.11 Applying Social Network Analysis Techniques to Software Engineering

This lecture presents an approach to the use of Social Network Analysis techniques in software engineering. Humans are by and large social beings by nature, and naturally form networks offline, the creation and maintenance of which are increasingly being supported online as well thanks to advances in supporting technology. In business settings, useful contacts are increasingly reported as being met outside of the firm, often through business networking events. Research reports that knowledge sharing from one's personal network has more of an impact on the success of a project than information obtained from knowledge management systems, and suggests that the relationships between individuals might matter more in this context than the individuals' attributes themselves. Social Network Analysis permits the analysis of those links and relationships, interactions, and information flows. This lecture demonstrates how social networks paradigms can support designing IT-enabled business processes, presents the steps involved in the process, and provides as an example an order fulfillment process where relevant data was collected before and one month after the implementation of IT in the business process.

2.1.12 An Introduction to the Analysis of Computational Systems through Abstract Interpretation

This lecture pertains to program analysis techniques applicable to several levels of Service-Oriented Applications, for instance to ensure their properties (e.g. correctness, robustness) or to analyze and synthesize service orchestration: checking functional properties, resource consumption etc.

The concept of abstract interpretation is introduced, and a comparison with other program analysis methods presented. An overview of abstract interpretation approaches and issues are then provided as well as definitions of the abstract meaning of a program and an introduction of abstract domains. finally, abstract interpretation for logic programs - which aim at expressing programming concepts and constructs in terms of logical statements suitable for formal and automated reasoning - is presented along with a case study.

2.1.13 Service-Centric Systems and Requirements Engineering

This mini-tutorial introduces web services and service- centric systems and explores their impact on requirements engineering processes, techniques and tools. Web services, service-centric systems and

service-oriented architectures are changing the ways in which we develop software systems, however the changes needed to requirements processes and techniques have not been explored widely.
In the mini-tutorial we will introduce web services and service-oriented architectures and current trends in this field, then introduce new tools and techniques with which to engineer requirements for service-centric systems developed in the EU-funded SeCSE project (secse.eng.it). These include tools and techniques for specifying and publishing web services with functional and quality features, discovering web services compliant with early requirements, specifying service-level agreements from requirements, and service monitors based on requirements.

2.1.14 Service-based software development

In order to develop Service-based Applications, it is necessary to extend existing software development practices and approaches with new processes, methods, and tools to permit the discovery and composition of services that can fulfill the functional and non-functional requirements of the SBA. This module presents a framework for service-based software development; it provides an overview of the framework, presents its goal and processes, and details a prescribed query language to specify service discovery requests. Algorithms for similarity analysis and distance measures are presented, and a ConferenceTravel SBA example is used to illustrate the concepts introduced.

2.1.15 A framework for proactive dynamic service discovery

Replacing services composing an SBA is sometime necessary to ensure its continued delivery of good service. Such run-time replacements can permit to avoid unavailable or malfunctioning services in the SBA; address changes in the he structure, functionality, quality, or context of the participating service or the system; or stem from arising opportunity to take advantage of a comparatively better service made available. This module presents an overview of a service discovery framework and of its architecture. Service discovery queries and a corresponding query language are presented and illustrated; different execution modes for dynamic service discovery are then introduced. Finally, an evaluation of the framework is provided.

2.2 Business process Management

2.2.1 Web Services Business Process Execution Language (WS-BPEL)

The BPEL overview lecture deals with the standard service composition language, WS-BPEL. It describes the major usages of the language and explains its relation to the Web service stack. Then based on an example business process, all language features are explained in detail: activity types, partner interaction, correlation, variable usage, and advanced concepts such as dead-path elimination, scopes, fault and compensation handling, abstract processes etc.

The subsequent BPEL and Aspects lecture require an understanding of aspect-oriented programming (AOP); it deals with the flexibility of service compositions. It introduces a classification of process flexibility approaches and presents a specific approach which utilizes aspect-oriented programming techniques for increasing the flexibility of WS-BPEL service compositions.

2.2.2 Agile Service Networks

Many of today’s service economies use ICT enabled around-the-globe networks for communication, contract management and monitoring, and/or mechanisms for trust building in global networks. Agile Service Networks comprise large numbers of service interactions that typically span organizations and geographical locations. This lecture introduces the notion of service economies, service value, and concepts of service systems as dynamic, human-centered value-cocreation systems. It then presents an overview of existing research on service value system. Finally, it presents an approach to define the value of a value network and mappings to processes, people and services.

The course's target audience are students in e-business, business transformation, software engineering, service engineering and computer science.

2.3 Infrastructure and Middleware

2.3.1 Introduction to Grid computing

A Grid is a heterogeneous collection of distributed computers, devices and services that are connected by a wide-area network, can dynamically join and leave the Grid, and are accessible on-demand by a set of users. This set of 5 lectures and presentations introduces concepts and principles relating to Grid Computing and the relationship between grids and SOA.

- **Lecture 1**: This set of slides gives an introduction to the notion of grid computing, motivations for their use, examples of application areas, and presents grid infrastructure and applications. Grid as SOA implementations and grid realized as a service infrastructure are also explained.
- **Lecture 2**: This presentation introduces the EGEE grid infrastructure and its middleware, gLite as well as related high level tools. Access, example usage scenario, main components and overview of the user community are explained as well as the concept of grid execution environments and portals.
- **Lecture 3**: This presentation exemplifies how grid service compositions, often called grid workflows, can be created and executed in grid portals. The required steps are demonstrated through a widespread portal solution called the P-GRADE Portal.
- **Lecture 4**: This presentation surveys the concept and capabilities of the WS-Resource Framework of Globus Toolkit 4. The WS-Resource Framework defines conventions for managing state so that applications discover, inspect, and interact with stateful resources in standard and interoperable way while maintaining the context of established Web services standards. Web service states, their modelling, and examples are presented.
- **Lecture 5**: finally, this presentation introduces the notion of executing legacy applications as grid services. This concept is realized by the GEMLCA portal (Grid Execution Management for Legacy Code Applications).

3 Summary and conclusion

This deliverable reported on the learning material collected from S-Cube partner for use in the SoE 1.1 Work package as part of the Virtual Campus (also see http://vc.infosys.tuwien.ac.at/), and provided a description of each collected item, either individually or as part of a higher level module. Overall, the material collected so far covers much of the service lifecycle and provides base elements on Service-oriented Systems principles and technologies as well as more specialised elements. Some areas of SBA lifecycle however are not currently covered by the available elements, namely: monitoring, composition, adaptation and deployment, and provisioning. These gaps are expected to be addressed during forthcoming collections of material as planned in the Description of Work for later deliverables (e.g. CD-SoE-1.1.5: "Completed learning unit collection").