Grant Agreement N° 215483

Title:	*Initial learning programme and module collection*
Authors:	*Lero, UniDue, UOC, VUA, TUW, USTUTT, INRIA, SZTAKI, Tilburg, CITY, UPM, TUW, Polimi*
Editor:	*Angela Kounkou (CITY)*
Reviewers:	*Philipp Leitner (TUW)*
	Christos Nikolaou (UOC)
Identifier:	*CD-SoE-1.1.3*
Type:	*Deliverable*
Version:	*1*
Date:	*15 March 2010*
Status:	*Final*
Class:	*Internal*

Management Summary

This document reports on the initial set of learning material collected as part of S-Cube's Spreading of Excellence (SoE) activities. WP-SoE 1.1 ("Virtual Campus") aims among others to make widely available various lectures and training programmes on research and applications of software services, including a joint Masters and PhD programme.

In this deliverable and as a step towards this goal, we collected learning material from S-Cube partners, undertook an initial classification of the items, and provided a description of the material currently available. We further provided a preliminary overview of joint curricula derived from the available material, and suggested gaps in the current collection that may be addressed by subsequent additions as part of upcoming S-Cube deliverables.

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

List of Tables

1 Introduction

In this deliverable, we report on learning material collected from S-Cube partners and their fit into the overall goal to structure and provide learning programmes through an S-Cube Virtual Campus. This section presents the context for learning material collection with regards to its aims and relations with other workpackages (section 1.1). It further sets the context of joint higher education curricula for European institutions, and introduces a similar joint curriculum initiative in the field of services science (section 1.2).

The remainder of this document is organized as follows. Section 2 presents an overview of the planned S-Cube joint curricula; section 3 presents the initial set of learning material available; section 4 provides a gap analysis of the collected material with regard to S-Cube research; and section 5 summarizes and concludes this report.

1.1 Workpackage vision

Spreading of Excellence (SoE) activities, as described in the S-Cube Description of Work[1], aim among other to disseminate S-Cube research results in the wider academic and scientific communities and to provide training for researchers. These activities are shared between two work packages: SoE 1.1 and SoE 1.2. The purpose of WP-SoE-1.1 ("Virtual Campus") more specifically is to:

- Provide an infrastructure to support training for academic and industrial research
- Provide a virtual community network facilitating communication and cooperation between students and researchers from various countries and backgrounds;
- Provide exposure to different academic approaches to Software Services and Systems research
- Set up Software Services and Systems competence centres providing access to service-related expertise, technologies and knowledge to increase the take-up of service technologies
- Collect and consolidate course material on Software Services and Systems research and applications
- Structure and develop training and education programmes on the topic of service-oriented computing, including joint Master's and PhD programs;

This deliverable relates to the latter two points. It contributes to the structuring and development of training and education programmes by collecting, reviewing and reporting on available material shared by S-Cube partners. It further provides input for the following deliverables that will jointly describe courses and the modules comprising them:

- CD-SoE-1.1.5 ("Completed learning unit collection") comprising a detailed report describing jointly defined courses and a complete description of modules per each course;
- CD-SoE-1.1.7 ("Initial Masters programme") detailing the steps towards establishing a joint European Master's programme;
- CD-SoE-1.1.8 ("Initial PhD programme") detailing the steps towards establishing a joint European Ph.D. program.

Additionally, it must be noted that the learning modules collected and outlined here are described in more detail in deliverable SoE 1.2.6 ("Creation of course registries").

1.2 Joint European Higher Education programmes

As evidenced by the Bologna Process and the Erasmus Mundus programmes, there currently is a drive towards cross- University and cross-country cooperation in European Higher academic institutions and beyond. The Bologna Process[2] aims to *"create a European Higher Education Area (EHEA) and to*

[1] http://www.s-cube-network.eu/
[2] http://www.ond.vlaanderen.be/hogeronderwijs/bologna/

promote the European system of higher education worldwide". Institutions are encouraged to establish co-operation and partnership links, notably by sharing knowledge for curriculum development and for the establishment of joint European educational degrees. The Erasmus Mundus programme[3] 2009-2013 operates in the continuity of its first phase (2004-2008), and aims among others for "*the enhancement of quality in European higher education*" and "*the promotion of the European Union as a centre of excellence in learning around the world"*. In this optic, it supports higher education institutions involved in projects enhancing the profile and visibility of European higher education internationally.

Joint European education programmes involve two or more higher education partners in different countries contributing their knowledge, teaching and supervision capacities. They are compliant with the Bologna declaration and with the national laws of each participating institutions as well as European standards for quality assurance. They typically include periods of study abroad in partner institutions, and lead to the award of one or more legally recognized degrees upon completion of the programme. One such joint programme in the field of service science is the joint International Master in Service Engineering (IMSE[4]) which, incidentally, involves members of the S-Cube consortium – namely the University of Stuttgart, The University of Crete, and Tilburg University.

2 S-Cube joint curricula vision

In partnership with the consortium's universities, S-Cube aims to derive from existing curricula and student exchange programs a common graduate curriculum that will provide students from different scientific disciplines with a foundation in Software Services and Systems. Interdisciplinary, European PhD and Master's programmes influenced by S-Cube research activities will be developed to offer both introductory and advanced courses for enrolled students. It is envisaged that students from other programmes will also be able to select elective modules from the joint curricula for their own course of study. The mobility of Master and PhD students between the involved universities will be promoted, with each student being appointed a head supervisor from one of S-Cube's partner institutions and being offered the possibility to spend some of their research time in one of the network's universities.

2.1 Motivation

In the past few years, service science has emerged as a major research topic and has encountered a rapid growth in its uptake and application in a variety of domains. This state of facts is fostered by several factors, for instance the natural evolution of computing paradigms towards software reuse; the backing of major IT providers; but also economical trends linked to the increasing expansion of companies from a local to a global scale thanks to the evolution of existing technology and to the availability of new technology. As a result, there is a demand for highly skilled service professionals to support the drive towards service-oriented systems and to research, engineer and manage services and service networks. Joint curricula axed around service-oriented computing will contribute to addressing the need for research and education on the topic.

2.2 Objectives

The overall objective for the joint curricula as described in the S-Cube Description of Work is to foster the alignment and integration of European competence and knowledge, and to educate young researchers while removing obstacles to the exchange of ideas and integration of research that stem from differences in institutions and in background disciplines. More specifically, the aim of the joint curricula is to:

- contribute to a durable base and infrastructure for European service-based systems research and education

[3] http://eacea.ec.europa.eu/erasmus_mundus/programme/about_erasmus_mundus_en.php
[4] http://www.erasmusmundus-imse.eu/

- provide a solid background, a common view and an understanding of the main service-based systems technologies and applications for the next generation of SBA researchers
- engage with and integrate diverse research communities, including Grid Computing, Service Oriented Computing and Software Engineering
- ensure that young researchers have the opportunity to work with S-Cube beneficiaries on Software Services and Systems

2.3 Target audience and learning outcomes

The joint curricula target Master students, PhD Students, practitioners with relevant experience, and young researchers across Europe with a background in Business Administration, Computer Science, Information Systems, or other related areas (e.g. Software Engineering, Grid Computing) wishing to gain a thorough grounding in Service-Oriented Computing and/or to specialize in related topics.

The overall learning outcomes aimed for are a comprehensive, up-to-date knowledge of service engineering theory and practice; the acquisition of skills applicable to related academic and industrial professions; and the acquisition of experiences and perspectives from a European rather than national viewpoint. Upon completion of their curriculum of study, students are expected to be competent in services and service networks engineering, and aware of the latest developments in the rapidly evolving domain of service-oriented computing.

2.4 Structural overview and challenges

In terms of taught modules, the joint curricula is foreseen to be organised around compulsory, core modules (here defined as offering relevant basic knowledge of service-oriented computing) and a set of elective, more specialised modules focused on specific parts of the programme of study. All S-Cube partners will contribute material to the joint programme of study; anticipated challenges in the definition and implementation of the joint curricula will include agreeing on a common specification for each programme, integrating the contributed material, and ensuring consistency across the participating institutions. Other practical challenges foreseen in the implementation of the joint programmes in all partner institutions relate to their integration with existing work practices due to variations in programme structures and lengths, in holiday scheduling and academic year length, and in the course assessment systems in effect across institutions and countries. Finally, legal questions related to the recognition of the joint programmes both on a national and European scale are anticipated.

2.5 Topics covered

Although other programmes of study such as the before-mentioned IMSE set successful precedents of curricula axed around service sciences, their proposed list of topics does not dictate this deliverable's suggested list of topics and classification for the contributed learning material. The broad list of topics considered here is driven first and foremost by S-Cube's research agenda and by the partners' expertise. It is however expected that the upcoming structuring of the study programmes (e.g. CD-SoE-1.1.7) duly examine and emulate such courses' structure if and where appropriate. Assuming the legal and logistic aspects of implementing such joint curricula have been overcome, the programmes of study will encompass topics from all of S-Cube's Joint Research Activities work packages:

- **JRA-1: Engineering and Adaptation Methodologies for Service-based Systems**, which is aimed at devising an integrated set of principles, techniques and methodologies for: engineering, adapting and monitoring hybrid service-based applications, and guaranteeing end-to-end quality provision and SLA conformance. JRA-1 comprises the following workpackages:
  - WP-JRA-1.1 (engineering principles, techniques and methodologies for hybrid, service-based applications) which defines principles, techniques and methodologies for engineering hybrid service-based applications exploiting knowledge from fields outside of service-oriented systems engineering

- WP-JRA-1.2 (adaptation and monitoring principles, techniques and methodologies for service-based applications) which defines novel principles, techniques and methodologies for cross-layer monitoring of service-based systems.
- WP-JRA-1.3 (end-to-end quality provision and SLA conformance) which defines principles, techniques and methodologies for specifying, negotiating and assuring end-to-end quality provision and SLA conformance with respect to quality characteristics across functional layers and across the chain of service providers and consumers.

- **JRA-2: Realisation Mechanisms for Service-based Systems** is concerned with functional SBA layers: business process management (BPM), service composition and coordination (SCC), and service infrastructure (SI). It aims to devise novel mechanisms achieving the technological innovation required for the functional SBA layers, and to develop seamless sophisticated interoperation mechanisms to guarantee that all three layers operate in harmony. JRA-2 comprises the following workpackages:
  - WP-JRA-2.1 (business process management) which provides concepts for service implementation from business models, business transactions monitoring, and better collaboration and decision-making within integrated Agile Service Networks.
  - WP-JRA-2.2 (Adaptable Coordinated Service Compositions) which investigates service composition and coordination for adaptable service-enabled business processes in multiple domains, taking into account stakeholders needs, QoS characteristics of processes and key performance indicators.
  - WP-JRA-2.3 (Self-* Service Infrastructure and Service Discovery Support), which defines policies, monitoring and redeployment techniques for self-adaptive and self-healing services, and develops various registry support mechanisms.

Overall the topics covered in these workpackages, many of which are cross-cutting, can be classified as follows: *Engineering*, including all phases of services and SBA lifecycle (design time – e.g. requirements elicitation and design, and runtime - e.g. adaptation and monitoring); *Business process Management*, including Business concepts, Business process modeling, integration, execution and analysis; and *Infrastructure and Middleware*, including Grid computing, infrastructure and network concepts, and self-*. This initial classification is used in this deliverable to present the collected learning modules, and will be refined as the module collection continues and the programmes are structured for upcoming SoE deliverables.

3 Initial learning modules collected

The collected course materials pertaining to Software Services and Systems are documented here using templates adapted from SoE 1.1.2 (“Organisational structure for virtual campus”) that succinctly describe the module, its aims and its pre-requisites (where existing). More detailed specifications are available from deliverable SoE 1.2.6 (“Creation of course registries”). Contributions are classified according to their broad topic area, and will be refined during the development of the joint curricula; the template proposed in SoE 1.1.2 will be implemented if and where appropriate for the description of fully fledged courses.

3.1 Engineering

This section covers all phases of services and SBAs lifecycle: requirements engineering; design; construction and quality assurance; deployment and provisioning; operation, management and quality assurance; and adaptation. It additionally include the related topics of HCI and context factors.

Table 1: Advanced internet Computing

Learning module	Advanced Internet Computing
Author	TUW
Related WP	WP-JRA-1.1
Description	The module discusses theoretical foundations, technologies, architectures, standards and examples of recent developments regarding Internet Computing and their applications, focusing on Service-oriented Computing and Web services. Topics covered include SOC and SOA, 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, WSC)
Prerequisites	Good programming knowledge in Java
Knowledge model terms	Service; Software Service; Service-Based Application; Service Registry; Service Provider; Web Service; Stateless Service; Stateful Service; Service Deployment; Service Level Agreement; Service Orchestration; Service Choreography; Business Activity

Table 2: Service Oriented Design

Learning module	Service Oriented Design
Author	VUA
Related WP	WP-JRA-1.1
Description	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. Topics covered include software services and their key aspects, concepts of SOA, service discovery, service composition, service modelling, QoS and SLAs, and key differences between system development and service oriented development.
Prerequisites	N/A
Knowledge model terms	Service; Software Service; Service Design; Service Discovery; Service Composition; Enterprise Service Bus; Service Orchestration; Service Choreography; Service Composition; Service Analysis

Table 3: Introduction to Model-Driven Engineering

Learning module	Introduction to Model-Driven Engineering
Author	INRIA
Related WP	WP-JRA-1.1
Description	This module introduces modeling as a way to master the complexity of modern software development and presents concepts and principles of model-driven engineering. Topics introduced include: UML and Model Driven Engineering, object-oriented meta-languages, model transformation techniques, software product lines (SPL), and aspect oriented model driven engineering.
Prerequisites	N/A
Knowledge model terms	Model-Driven Service Composition

Table 4: Web Services in the Semantic Web

Learning module	Web Services in the Semantic Web
Author	UOC
Related WP	WP-JRA-1.1
Description	This module describe technologies to support a machine-interpretable web,

	and mainly deals with the issue of describing Semantic Web Services. Topics presented include the concept of ontologies as a tool for the creation of enriched WS descriptions and ontology technologies.
Prerequisites	N/A
Knowledge model terms	Semantic Web Services; Web Service; Service Description

Table 5: A QoS Ontology and a Semantic QoS Metric Matching Algorithm

Learning module	A QoS Ontology and a Semantic QoS Metric Matching Algorithm
Author	UOC
Related WP	WP-JRA-1.1
Description	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, OWL-Q, one of the richest of such QoS ontologies, is presented. 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.
Prerequisites	Basic knowledge of standard WS and Semantic Web technologies
Knowledge model terms	Software Service; Web Service; Semantic Web Services; Quality of Service Characteristic

Table 6: QoS-Based Web Service Discovery

Learning module	QoS-Based Web Service Discovery
Author	UOC
Related WP	WP-JRA-1.1
Description	In this presentation, novel algorithms for QoS-based Web Service matchmaking and selection are analyzed, and an empirical evaluation conducted on specific implementations of these algorithms, that use different constraint solving techniques, is presented.
Prerequisites	Basic knowledge of standard WS technologies
Knowledge model terms	Software Service; Web Service; Service Discovery; Quality of Service Characteristic

Table 7: Software Quality & Processes

Learning module	Software Quality & Processes
Author	Lero
Related WP	WP-JRA-1.3
Description	This module presents concepts and principles relating to software quality and processes. Topics introduced include: product quality, focusing specifically on a framework for Total Quality Management; advantages and disadvantages of implementing and improving processes within the software organization; implementation of process thinking, process focus and process discipline; implementation of the effective software process; and software process quality measurements.
Prerequisites	N/A
Knowledge model terms	Process; Software Process; Software process model; Quality of Service Dimension; Capability Maturity Model; Capability Maturity Model Integration; Key Performance Indicator

Table 8: Quality Assurance for Service-based Systems: From Software Engineering to Service Engineering

Learning module	Quality Assurance for Service-based Systems: From Software Engineering to Service Engineering
Author	UniDue

Related WP	WP-JRA-1.3
Description	This module motivates 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: Testing, Monitoring, and Static Analysis.
Prerequisites	Basic background in engineering processes and techniques
Knowledge model terms	Software Quality Assurance; Software Service; Service-Based Application; Service; Failure; Fault; Error; Static Analysis; Testing; Monitoring

Table 9: Human Provided Services in Mixed Service-Oriented Systems

Learning module	Human Provided Services in Mixed Service-Oriented Systems
Author	TUW
Related WP	WP-JRA-1.1
Description	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.
Prerequisites	Basic knowledge of standard WS technologies
Knowledge model terms	Service; Software Service; Service-Based Application; Web Service; Human Computer Interaction; Activity; Context

Table 10: Applying Social Network Analysis Techniques to Software Engineering

Learning module	Applying Social Network Analysis Techniques to Software Engineering
Author	Lero
Related WP	WP-JRA-2.1
Description	This lecture presents an approach to the use of Social Network Analysis techniques in software engineering and demonstrates how social networks paradigms can support designing IT-enabled business processes and the steps involved in the process.
Prerequisites	N/A
Knowledge model terms	Business Process; Social Network Analysis; Business Process Measurement; Business Process Analysis, Monitoring and Auditing; Key Performance Indicator

Table 11: Introduction to the Analysis of Computational Systems through Abstract Interpretation

Learning module	An Introduction to the Analysis of Computational Systems through Abstract Interpretation
Author	UPM
Related WP	WP-JRA-1.1
Description	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 topics presented include: concept of abstract interpretation; program analysis methods; abstract interpretation approaches and issues; abstract domains; abstract interpretation for logic programs.
Prerequisites	N/A
Knowledge model terms	Service analysis

Table 12: Service-Centric Systems and Requirements Engineering

Learning module	Service-Centric Systems and Requirements Engineering
Author	CITY
Related WP	WP-JRA-1.1
Description	This tutorial introduces web services and service- centric systems and explores their impact on requirements engineering processes, techniques and tools. Topics presented include: Web services, service-centric systems, service-oriented architectures, requirements engineering tools and techniques, and 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.
Prerequisites	N/A
Knowledge model terms	Service; Software Service; Service-Based Application; Web Service; Requirements Engineering; Requirement; Requirements Analysis

Table 13: Quality of service optimization in web services

Learning module	Quality of service optimization in web services
Author	POLIMI
Related WP	WP-JRA1.3
Description	This module discusses techniques to evaluate quality in service compositions. It presents the motivating needs for quality in SOA, introduces the concepts of QoS optimization and quality-driven compositions, and provide an analysis of optimization techniques. Reoptimization criteria at run time are illustrated. Triggers for rebinding, possible causes for failure and their handling, and available tool support are presented.
Prerequisites	Knowledge of BPM
Knowledge model terms	Quality Definition, Negotiation and Assurance (KM-QA); Quality negotiation and agreement (contract establishment); Optimization; Process; Quality of Service Constraint; Rebinding; Cost Model

Table 14: Data quality in adaptive service systems

Learning module	Design of dependable service-based processes
Author	POLIMI
Related WP	WP-JRA1.1
Description	This module discusses service-based processes dependability and illustrates techniques for improving process dependability. A definition of data quality is provided along with causes of poor data quality, assessment and measurement. Service adaptation requirements are then introduced and an algorithm for selecting adaptation strategies at design and at run time is illustrated. Finally a methodology to improve Web Service dependability is presented.
Prerequisites	Knowledge of BPM
Knowledge model terms	Service Based Application Construction; Service Specification; Adaptation; Adaptable Service-Based Application; Adaptation Requirements and Objectives; Data-Related Quality; Evolution; Adaptation Strategy; Design for Adaptation; Dependability; Self-Healing; Process; Context; Cost Model

Table 15: Service-based software development

Learning module	Service-based software development
Author	CITY
Related WP	WP-JRA-1.1

Description	This module presents a framework for service-based software development. An overview of the framework introducing its goal, processes and prescribed use of a query language to specify service request is provided. Algorithms for similarity analysis and distance measures are presented, and a ConferenceTravel SBA example is used to illustrate the concepts introduced.
Prerequisites	Basic knowledge of standard WS technologies
Knowledge model terms	Service; Software Service; Service-Based Application; Service Based Application Construction; Service Discovery

Table 16: A framework for proactive dynamic service discovery

Learning module	A framework for proactive dynamic service discovery
Author	CITY
Related WP	WP-JRA-1.1
Description	This module provides a motivation for proactive service discovery and presents an overview of a service discovery framework and 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.
Prerequisites	Basic knowledge of standard WS technologies
Knowledge model terms	Service; Software Service; Service-Based Application; Service Discovery

3.2 Business Process Management

This section focuses on Business Process definition, deployment, execution, monitoring, analysis and evolution as well as related enabling technologies (e.g. BPEL).

Table 17: Web Services Business Process Execution Language (WS-BPEL)

Learning module	Web Services Business Process Execution Language (WS-BPEL)
Author	USTUTT
Related WP	WP-JRA-2.2
Description	This lecture deals with the standard service composition language, WS-BPEL, and describes the major usages of the language its relation to the Web service stack. 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.
Prerequisites	Basic knowledge of standard WS technologies
Knowledge model terms	Web Service; Service Composition; Workflow; Process; Business Process; Fault; Activity

Table 18: Aspects and BPEL

Learning module	Aspects and BPEL
Author	USTUTT
Related WP	WP-JRA-2.2
Description	This presentation deals with the flexibility of service compositions. It introduces a classification of process flexibility approaches and presents then a specific approach which utilizes aspect-oriented programming techniques for increasing the flexibility of WS-BPEL service compositions.
Prerequisites	Basic background in Web Services, BPEL, and aspect-oriented programming (AOP).
Knowledge model terms	Web Service; Service Composition; Workflow; Process

Table 19: Agile Service Networks

Learning module	Agile Service Networks
Author	UOC
Related WP	WP-JRA-2.1
Description	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.
Prerequisites	N/A
Knowledge model terms	Agile Service Network; Value Network

3.3 Infrastructure and Middleware

This section focuses on the technical foundations – software and hardware - on which research in the other JRA workpackages is based and includes self-* functionalities and notions of Grid computing.

Table 20: Introduction to Grid computing

Learning module	Introduction to Grid computing
Author	SZTAKI
Related WP	WP-JRA-2.3
Description	This module introduces concepts and principles of Grid Computing and the relationship between grids and SOA. The topics presented include: notion of grid computing, motivations for use, infrastructure and middleware, grid service compositions creation and execution, and execution of legacy applications as grid services.
Prerequisites	N/A
Knowledge model terms	Grid; Grid Workflow; Stateless Service; Stateful Service

4 Gap analysis

We collected learning material from 13 partners who provided items from various domains, encompassing a wide range of topics related to Service Oriented Computing. A comparison of the initial set of material against the preliminary list of topics to be covered (outlined in section 2.5) reveals unbalances in the material currently available, and a number of topics addressed in S-Cube's joint research work packages are not currently presented in the initial set of collected learning modules.

While WP- JRA-1.1 accounts for 13 learning items out of 20, the other workpackages are clearly under-represented, and WP-JRA-1.2 is not featured at all. Aside from workpackage classification, specific topics can also be identified as lacking from the initial set collected.

- In *Engineering*, monitoring and adaptation material are missing, as are items on deployment and provisioning, HCI, composition and coordination models, approaches and verification, and operation and management.
- The *Business Process Management* subsection does not currently include material on Business Processes modelling, technical infrastructure, and integration.
- Finally, the *infrastructure and middleware* ***section*** currently lacks material on service networks, their engineering, and enterprise service bus.

It is recommended that the next and final collection of learning material (CD-SoE-1.1.5, due for month 33) specifically seeks to ensure that contributed material addresses the gaps highlighted here in order for the planned programmes to reach their educational objectives.

5 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. Overall, the material collected so far spans much of the service lifecycle and provides elements for solid foundations on Service-oriented Systems' architecture, design and execution while providing context on the environments in which these systems are expected to be engineered and operated. Concepts, principles, approaches and techniques for service requirements elicitation, service description, discovery and quality assurance were contributed as well as insights into engineering, analysis and design techniques and the human actors of such systems. It is however possible to identify areas of SBA lifecycle that are not currently covered by the available elements, namely: monitoring, deployment, and provisioning. Concepts of BPM, HCI and service networks infrastructure were also found to be lacking. These gaps are expected to be addressed during the structuring of learning programmes and subsequent collections of material as planned in the Description of Work for upcoming deliverables (e.g. CD-SoE-1.1.5: "Completed learning unit collection").