Grant Agreement N° 215483

Title:	*Results of the Second Validation*
Authors:	*UniDue, POLIMI, CITY, FBK*
Editors:	*Eric Schmieders (UniDue), Andreas Metzger (UniDue)*
Reviewers:	*Michael Parkin (Tilburg)*
	Ita Richardson (Lero-UL)
Identifier:	*CD-IA-3.2.4*
Type:	*Deliverable*
Version:	*1*
Date:	*14 March 2011*
Status:	*Final*
Class:	*External*

Management Summary

In this document, Deliverable CD-IA-3.2.4 "Results of the Second Validation" we report the validation of the integration of the IRF building blocks, i.e., the results of task T-IA-3.2.1.
This deliverable describes the vision and strategy of the work package WP-IA-3.2 (using the Description of Work Amendment #4 as basis). A description of the work package's roadmap until the end of the S-Cube project is given, including the validation strategy and the interaction with other WPs. The validation method is introduced and applied to the IRF. The results of this application are listed. Finally, the summarized results and the conclusions are presented.

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 Workpackage Vision and Objectives

As defined in the Description of Work[1] the objectives and tasks of WP-IA-3.2 are:

- T-IA-3.2.3: The IRF validation of building blocks is checked by an *empirical evaluation* of the building blocks of the integration framework. The empirical evaluation will use, for instance, demonstrators, experiments, case studies and other appropriate empirical research methods. This objective will be supported by providing access to evaluation setups and results via the IRF.
- T-IA-3.2.1: The IRF validation of the *integration* of the building blocks in the integration framework through suitable high-level scenarios along the service life cycle. The definition of the validation scenarios will start with a collection and analysis of existing scenarios through a systematic survey. Next existing scenarios may be analyzed and extended, or in case appropriate ones are lacking, new scenarios may be devised. The ultimate goal of the validation is to revise and improve the integrated research framework. For this reason it will be conducted iteratively in different phases of the duration of the activity. Furthermore, the task comprises the collection of stakeholders associated to the high-level scenarios.

Together, those tasks will contribute to the consolidation of the IRF (IA-3.1). The collected stakeholders are provided to JRA-1.1. Work package JRA-1.1 can use the stakeholders to advance the initial set of stakeholders in terms of their usage in the life-cycle.

1.1 Empirical Validation of the Research Results (T-IA-3.2.3)

The aim of this task is a partial evaluation of the building blocks of the integration framework. A specific empirical evaluation may focus on a single building block or may cross-cut multiple building blocks. Methods for the empirical evaluations can include among others, laboratory and field experiments or case studies. This might require experiment specific coupling of tools and infrastructures. In addition, this task aims to support the set-up of experiments. In addition, it provides a structured access to validation results (e.g., by linking from research questions in the IRF to papers that include the validations).
This task will:

- Set up empirical evaluations of (parts of) the research results within the integration framework.
- Support the organisation, implementation and execution of the evaluation activities.
- Analyse the results of the evaluation activities.
- Provide a structured assess to validation results

1.2 Validation of the Integration of the Building Blocks in the Integration Framework (T-IA-3.2.1)

This task aims to validate the integrated research framework (IRF). To this end, high-level scenarios will be employed to check that the relevant activities of the service life cycle are covered. Those high-level scenarios will be defined in close cooperation with WP-JRA-1.1.
The task will iteratively take place during the different phases of the network. At each iteration, the task will be organized in the following steps:

- Collection, definition, adaptation and extension of high-level validation scenarios along the service life-cycle. These scenarios will consider, for example,
    - selected application domains from the S-Cube case studies in WP-IA-2.2;

[1] Deliverable CD-IA-3.2.4 is based on the WP-IA-3.2 outline of the Description Of Work Amendment #4 which includes major improvements based on readjustment Mgt-1.3.1.

  - o typical stakeholders involved in SBA design and adaptation as defined in WP-JRA-1.1
- Collection and alignment of research outcomes with the IRF (focussing on the life cycle view).
- Application of the selected scenarios.
- Evaluation of results as a basis for framework improvement.

This workpackage will contribute the set of scenarios obtained to the S-Cube convergence knowledge model.

1.3 Validation Object – The IRF

As described in CD-IA-3.1.5, the IRF is reshaped based on the result of the internal verification. The research challenges and questions have been updated as well. For this reason, IA-3.1 concentrates mainly on the work done in the JRA work packages that reflect, by definition, the research issues studied in the S-Cube project. Operatively, each JRA-WP leader was in charge of analysing the research work performed in their work packages in the last year in order to identify the relevant areas of study (for more details on the updated research focus of the JRA WPs also see CD-Mgt-1.3.1). At the same time, research challenges and questions that they do not consider relevant due to lacks of work on that or difficulties to really deal with them, are candidates to be dropped from the IRF. In some other cases, the research challenges and questions are only refocused according to the results obtained in the last period.

The research challenges and research questions in the IRF are directly maintained by the JRA-WP leaders. The challenges and research questions are hence kept consistent to the performed research (as described in CD-IA-3.1.5). Thus, this deliverable focuses on the integration of the research results.

1.4 Relation with other Integration Workpackages

For the overall strategy in WP-IA-3.2 it is important to understand the inputs and outputs needed and, therefore, to understand the relations and dependencies with the other integration workpackages. These dependencies are depicted in **Fehler! Verweisquelle konnte nicht gefunden werden.** and include:

- *WP-IA-3.1 (Integration Framework: Baseline and Definition) – WP-IA-.3.2*: The most important relationship of WP-IA-3.2 is the one with WP-IA-3.1 since WP-IA-3.1 provides the main inputs to WP-IA-3.2 in form of the IRF and of its research questions and research results, which are to be validated. In turn, WP-IA-3.2 provides the relevant materials in terms of validation results, which either become part of the IRF (validation of the IRF elements) or lead to an improvement of the IRF (validation of the entire IRF).
- *WP-IA-2.2 (Alignment with European Industry Practices) – WP-IA-3.2*: WP-IA-3.2 uses the industrial case studies from WP-IA-2.2 to derive validation scenarios. These validation scenarios are in turn used for extending/refining the industrial case studies and pilot cases (cf. [1]).
- *WP-IA-1.1 (Convergence KnowledgeModel) – WP-IA-3.2*: The knowledge model provides the relevant glossary terms related to the validation results.
- *WP-JRA-1.1 (Engineering Principles, Techniques and Methodologies for Hybrid, Service-based Applications) – WP-IA-3.2*: In WP-IA-3.2 stakeholders are collected which are related to the high-level scenarios. The stakeholders are provided to JRA-1.1 in order to refine the stakeholders (collected in JRA-1.1).

1.5 Roadmap and Timeline of IA-3.2

The roadmap and the timeline of IA-3.2 are based on the new description of work (Amendment #4). Given the vision and strategy outlined before and the dependencies between the workpackages, the following timeline will be used for WP-IA-3.2 for the years 3 and 4 (cf. Figure 1).

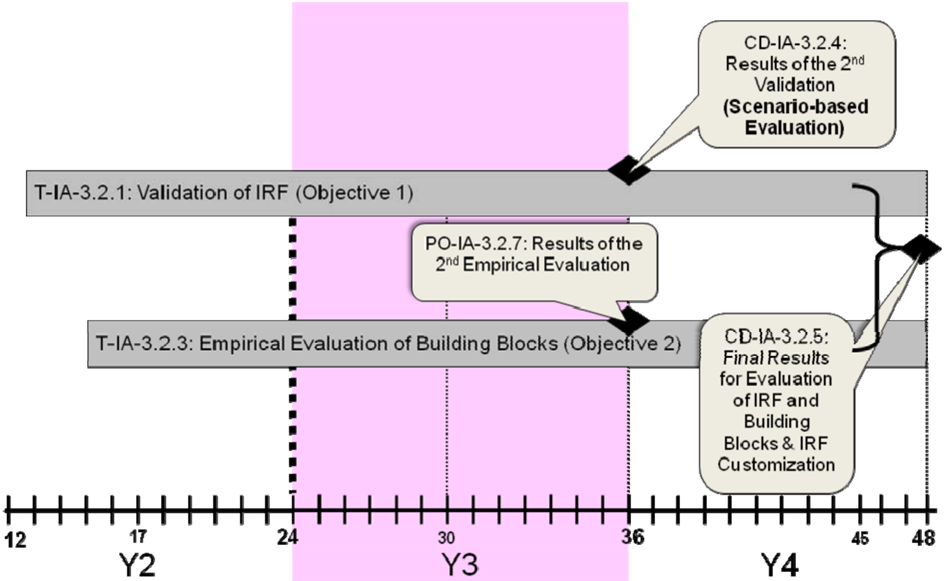

Figure 1: Timeline of IA-3.2

In the final year (Y4) the high-level scenario-driven validation is applied again (T-IA-3.2.1). A report regarding the validation status of the IRF elements will be produced (T-IA-3.2.3). This report will contain the elements, which are validated including the validation results. The report will also contain those elements, which were not yet validated during the S-Cube project.

2 Strategy for Validating the Integration of the IRF Building Blocks

The validation strategy for IA-3.2 was introduced in CD-IA-3.2.2 [55] and has been rationalized and streamlined based on the review recommendations and the readjustment of JPA (cf. Description of Work Amendment #4 and Mgt-1.3.1 [56]). Together with the re-focussing and rationalization of the IA-3.2 activities, the validation strategy for the *Validation of the Integration of the Building Blocks in the Integration Framework (T-IA-3.2.1)* is now described.

2.1.1 Use of High-Level Scenarios along the S-Cube Life Cycle

Based on the recommendations and the readjustment we reduced significantly the amount of purely "structural" assessments in the IRF in favor of a content-based validation. The validation assesses the integration of the novel research results generated by the S-Cube network. To achieve this, the former structural validation goals G1 to G6 (cf. CD-IA-3.2.2 [55]) have been rationalized and, instead, we exploit high-level scenarios based on the S-Cube life cycle to assess the integration of the research building blocks.

In this deliverable we use the S-Cube service life-cycle to understand the integration of the collected research results produced in the network and, beyond this, to determine the coverage of the IRF. Experiences within the S-Cube project and with external stakeholders have shown that the IRF life cycle view was ideal to understand, align, integrate and discuss the research outcomes. The service life cycle model envisioned by the S-Cube framework, shown in Figure 2, captures an iterative and continuous method for developing, implementing, and maintaining service compositions in which feedback is continuously cycled to and from phases in iterative steps of refinement and adaptations of all three layers of the technology stack. For further information about the life cycle, please consult PO-IA-3.1.4 [54].

In order to get an overview in IA-3.2 of the integration achieved between the building blocks, we asked the S-Cube members to report their research results, which integrate IRF building blocks. Every reported integration covers at least two (of the eight) life-cycle phases. This integration we consider as high-level scenario.

The high-level scenarios collected are already published or are planned to be published (see Section 3.1). Each high-level scenario refers to several results produced within S-Cube and aligned to the related phase of the life-cycle. The results addressed in the high-level scenarios are paper based, i.e. determined from publications by members of the S-Cube network.

2.1.1.1 High-Level-Scenario Example

To illustrate this strategy, let us assume a high-level scenario containing the three research results A, B and C. Result A relates to the *Operation & Management* phase, result B relates to the I*dentify Adaptation Need* phase and result C relates to the *Identify Adaptation Strategy* phase. Figure 2 depicts the alignment along the life cycle.

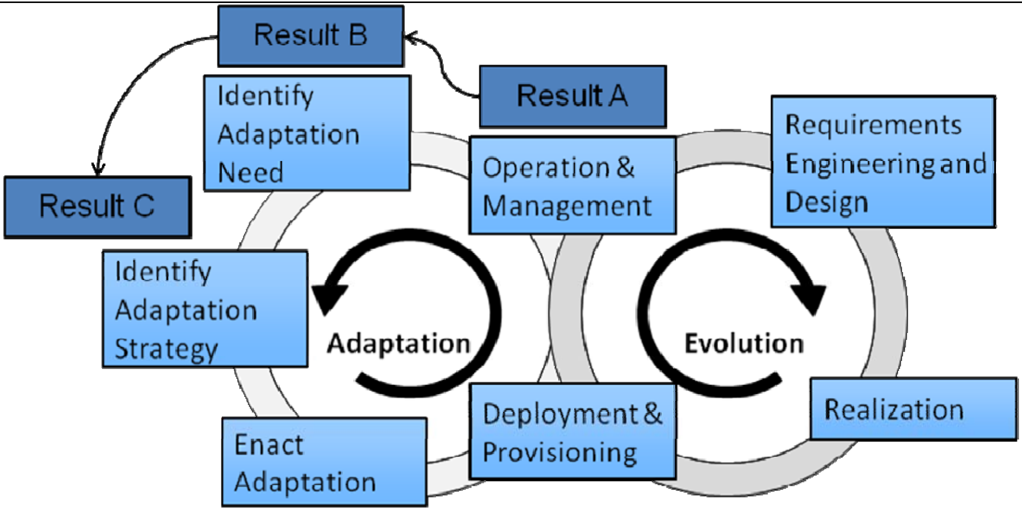

Figure 2: High-Level Scenarios

This high-level scenario reveals an integration of the three addressed phases. In consequence, once the high-level scenario is structured along the life cycle, we can identify life cycle phases which are not integrated. Furthermore, the alignment allows a content based validation of the integration performed, as supported by the last section of the template used for collecting the high-level scenarios. In order to extend the IRF coverage in terms of the three exiting layers (BPM, SCC and SI) the research results are also related to the relevant layers. By applying this methodology to every scenario collected from S-Cube publications, we can provide an overview of the integration of research results performed within S-Cube and the coverage of the IRF. Based on this overview, actions can be identified in order to improve the research integration during the last year of S-Cube, Y4.

2.1.1.2 Methodology

We collect the high-level scenarios by using a template. The template comprises three sections: “Detailed Scenario Description”, “S-Cube Life Cycle Coverage” and “Achieved Integration”. The template allows structuring the results along the life cycle.
The collected integration work has its origin in papers produced by S-Cube members (i.e. the work presented in Sections 3.1.1, 3.1.2 and 3.1.3) and in the ongoing integration work of work package JRA-1.2 presented in SoE-1.2.5 [57] (i.e. Sections 3.1.4, 3.1.5 and 3.1.6). In the former case we asked the authors to fill the template. In the latter case we asked the leaders of the three ongoing integrations to fill the template.

In the “Detailed Scenario Description” section of the template each high-level scenario is described in detail, providing background information, e.g. the motivation for the integration. In the second section “S-Cube Life Cycle Coverage” a table provides a row for each life cycle phase. Based on this, the high-level scenario is structured along the life cycle declaring the related functional layer (cf. Table 1). In addition, we asked the authors of the high level scenario to term involved stakeholders (collected in PO-IA-3.1.4 [54]) for each addressed life-cycle phase.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design			
Construction			
Deployment & Provisioning			
Operation & Management			
Identify Adaptation Needs			
Identify Adaptation Strategy			
Enact Adaptation			

Table 1: Template for Structuring the Research along the S-Cube Service Life-Cycle

The table furnishes information about each life cycle phase, the result addressing this phase and the related stakeholders and functional layers. Finally, in the "Achieved integration" section, the achieved integration and the open issues for each high-level Scenario are summarized.

The validation results are documented in Section 3. Each of the Sections 3.1.1 to 3.1.1 comprises one high-level scenario.

2.1.2 Assignment of stakeholder types

During the second IRF validation, we collected stakeholder involvements from the high level scenario authors, with the template presented above (cf. Section 2.1.1). To achieve this, we provide the stakeholder types defined in PO-IA-3.1.4 [54] (cf. Figure 3) to the authors.

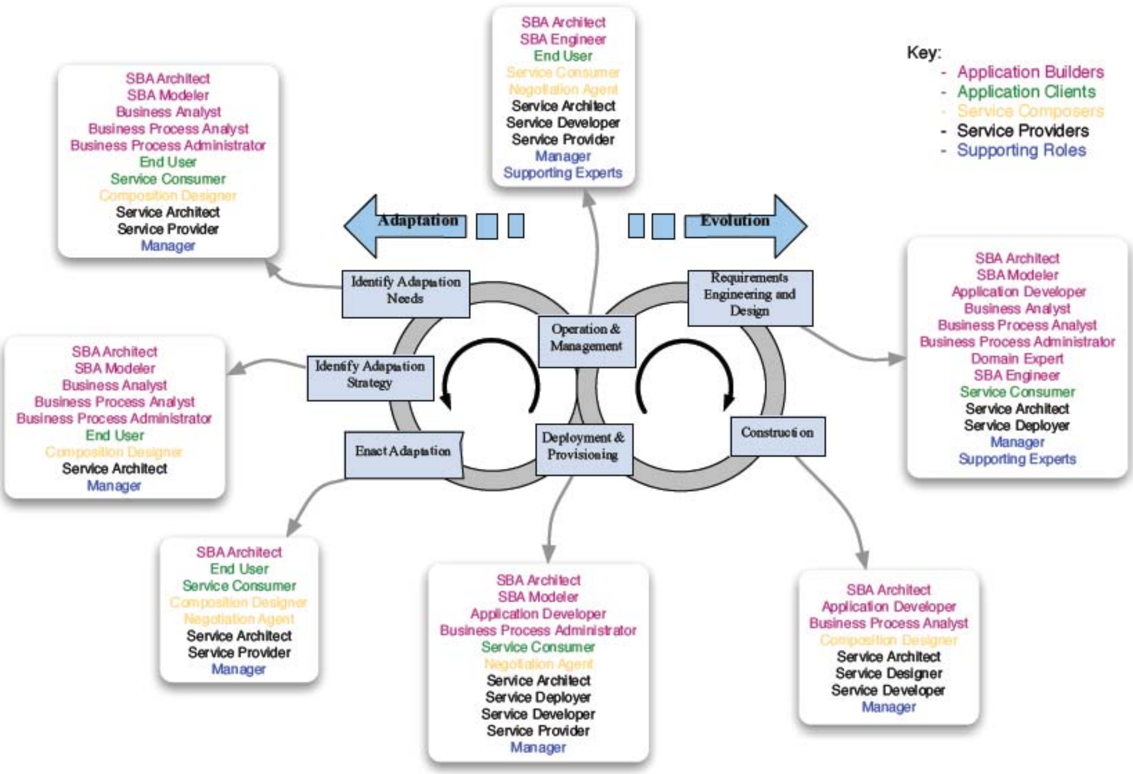

Figure 3: S-Cube Life cycle Phases with Stakeholder Types

The authors identified the stakeholders involved in their approach and completed the template accordingly. We will provide the results to work package JRA-1.1 supporting e.g. T-JRA-1.1.4 "Finalizing and evaluating the S-Cube lifecycle"[2]. We don't use the collected stakeholder involvements as input for the present validation.

[2] cf. Description Of Work Amendment #4

3 Validation Results

This section contains the second evaluation of the IRF. Given the timeline, strategy and vision outlined above, the deliverable focuses on validation of the IRF based on high-level scenarios.

3.1 Validation of the IRF Integration

The following 6 high-level scenarios have been validated (details are given in Sections 3.1.1 to 3.1.1) using the methodology described above:

- Pro-active Adaptation based on the S-Cube Service Life Cycle (published in [5], template filled by UniDue)
- Design for Adaptation of Service-Based Applications (published in [6], template filled by FBK)
- Real-time Online Interactive Applications (ROIA) (published in [16], template filled by POLIMI)
- Assumption-based Proactive Monitoring and Adaptation (planned to be published at SOCCER 2011, template filled by CITY)
- QoS-driven multi-layer monitoring and adaptation (planned to be published at SASO 2011, ECOWS 2011 or ICSOC 2011, template filled by POLIMI)
- Context-Aware Monitoring and Adaptation (planned to be published, template filled by POLIMI)

In Section 4 we give an overview of the achieved integration and are summarizing the key issues which should be addressed in future research and integration.

3.1.1 High-Level Scenario: "Pro-active Adaptation based on the S-Cube Service Life Cycle"

3.1.1.1 Detailed Scenario Description

The high-level scenario is based on the approach presented in [5]. This paper describes techniques to determine deviations from requirements based on monitored SLA violations. In this work, we demonstrate how the techniques for determining proactive adaptation work together across the various phase of the life-cycle and how they can be jointly applied in a meaningful way.

To assess automatically whether the application deviates from its requirements during operation and thus trigger an adaptation, functional and non-functional requirements need to be collected and formally expressed. Similarly to these requirements, the workflow of the SBA needs to be formalized to support automated checks. Following the same reasoning as in the requirements engineering phase, we suggest to formalize the workflow during the design phase already in order to reduce the risk of later corrections. During the realization phase, the quality levels (aka. Service level objectives) that have been negotiated and agreed upon with the service providers, are formalized. Before deploying the SBA, it is checked whether the workflow specification, under the given assumptions, satisfies the requirements. The operation and management phase comprises the execution and the monitoring of the individual services of the deployed SBA. In this phase it is checked, whether the requirements are still satisfied, although the assumptions have been violated. For example it might be the case that a slower response time of one service is compensated by a faster response time of a previous service, and consequently no adaptation is required. When the need for adapting an SBA is detected, the next step is to identify and apply an appropriate adaptation strategy among the ones that are available for the considered applications. Depending on the application, the adaptation strategies may range from service re-execution, over replacement of a single service or of the process fragment, over re-negotiation of quality properties, to changes in underlying infrastructure, etc. Note that the adaptation strategies should be designed with the application since some of them require the adoption of specific infrastructure or the implementation of additional components. To enact adaptation actions, the SBA or its execution platform should be appropriately instrumented.

3.1.1.2 S-Cube Service Life Cycle Coverage
The following table details the phases covered by the high-level scenario.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Usage of the Quality Meta Model (QMM) proposed [44]: Usage of ALBERT proposed [45] to formalize Requirements, proposed alternatives are the languages described in [22], [21] of the Monitoring Platforms Dynamo [48] and ASTRO [49]	SBA Architect, Application Developer, Business Analyst, Business Process Analyst, Domain Expert, SBA Engineer	BPM, SCC
Construction	Workflow formalization in BIR (input language for the BOGOR model checker) [45] Formalization of Assumptions [5]	SBA Architect, Application Developer	SCC
Deployment & Provisioning	Check if workflow specification together with assumptions fulfill the SBA requirements [33]	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	SCC
Operation & Management	Monitoring during runtime [21]	End User	BPM, SCC
Identify Adaptation Needs	Check if the workflow specification, the assumptions (of the not yet invoked services) and the monitoring data fulfill the requirements [33]	-	SCC
Identify Adaptation Strategy	Service substitution as adaptation strategy [38]	-	SCC
Enact Adaptation	Aspect orientated techniques used to enact adaptation [47]	-	SCC

Table 2: Detailed Coverage of the "Pro-active Adaptation based on the S-Cube Service Life Cycle"-Scenario

3.1.1.3 Achieved Integration
The high-level scenario enables a proactive adaptation in order to permanently meet the SBA requirements. The scenario covers the phases of the design time and runtime. By relating the research results, produced in S-Cube, to the each phase, the high-level scenario shows their integration. Furthermore, each of the three layers is addressed as depicted in the table.
The high-level scenario shows a good integration of the research results and covers all phases of the life cycle. The service infrastructure (SI) layer is not covered. The scenario has a strong focus on the service composition and coordination layer (SCC).

3.1.2 High-Level Scenario: "Design for Adaptation of Service-Based Applications"

3.1.2.1 Detailed Scenario Description

This High-level Scenario is based on the work published in [6]. Current methodologies for service-oriented applications are based on the results carried out in the fields of classical software and system engineering. Moreover, while almost all of the proposed approaches for life cycle, (noticeable are the proposals by SOUP (Service Oriented Unified Process) [7] or ASTRO [8] focusing on the possibility to monitor and intervene on SBAs in order to recovery from unwanted and unexpected behavior), assume human interventions, Linner et al. [9] propose a life cycle supporting self-adaptation of the service-based application even if they lack of an explicit guidelines for the design of adaptable service based applications. Various frameworks supporting adaptation have been defined in the literature, each of them addressing a specific issue. Some authors focus on triggering adaptation strategies as a consequence of a requirement violation [10], or for satisfying some application constraints [11]. Adaptation strategies could be specified by means of policies to manage the dynamism of the execution environment [12], [13], [14] or of the context of mobile service-based applications [15]. The high-level scenario is based on the approach presented in [6]. This paper proposes a design method for SBAs that targets the adaptation requirements of those applications and aims at overcoming the fragmentation in current approaches for SBA adaptation. The approach is based on a novel life cycle that considers adaptation as a first class concern and that covers the different facets of adaptation, both during the design phase and at run-time. Admittedly, this paper is just a first step towards our ultimate goal of defining a holistic design method for adaptable SBAs. Still, the effectiveness of such principles and guidelines is witnessed by their capability to capture the key aspects of adaptation in the different, heterogeneous real-world scenarios considered in this paper.

3.1.2.2 S-Cube Service Life Cycle Coverage

The following table details the phases covered by the high-level scenario.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	At the *requirements engineering and design* phase the adaptation and monitoring requirements are used to perform the design for adaptation and monitoring.	Adaptation Designer, Service Integrator	BPM,SCC
Construction	During SBA *construction*, together with the construction of the SBA, the corresponding monitors and the adaptation mechanisms are being realized.	SBA Architect, Application Developer, Adaptation Designer	BPM, SCC
Deployment & Provisioning	The *deployment* phase also involves the activities related to adaptation and monitoring: deployment of the adaptation and monitoring mechanisms and deployment time adaptation actions (e.g., binding).	SBA Architect, SBA Modeler, Application Developer, Adaptation Designer	BPM, SCC
Operation & Management	During the *operation and management* phase, the run-time monitoring is executed, using some designed properties, and help the SBA to detect relevant context and system changes. This phase instruments monitoring frameworks like ASTRO.	End User	SI
Identify Adaptation	Adaptation needs can be triggered from monitored events, adaptation	Monitoring Engine	SCC, SI

Needs	requirements or context conditions		
Identify Adaptation Strategy	For each adaptation need it is possible to define a set of *suitable strategies*. Each adaptation strategy can be characterized by its complexity and its functional and non functional properties. The identification of the most suitable strategy is supported by a *reasoner* that also bases its decisions on multiple criteria extracted from the current situation and from the knowledge obtained from previous adaptations and executions.	Adaptation Engine	SCC,BPM
Enact Adaptation	In this phase the *enactment of the adaptation strategy* selected is performed.	Adaptation Engine, SBA	SCC,BPM

Table 3: Detailed Coverage of the "Design for Adaptation of Service-Based Applications"-Scenario

3.1.2.3 Achieved Integration

As adaptation is the focus of the given high-level scenario (described with details in [6]), the factors for triggering adaptation are shown. The necessary artifacts are produced in the preliminary phases. Furthermore, the types of adaptation realization suitable for the scenarios, and the appropriate adaptation strategies and adaptation enactments are presented.
This high-level scenario covers all phases of the IRF life cycle and functional layers. It demonstrates a good integration of the building blocks with targeting runtime adaptation of SBAs.

3.1.3 High-Level Scenario: "Real-time Online Interactive Applications (ROIA)"

3.1.3.1 Detailed Scenario Description

The high-level scenario is based on the approach presented in [16]. This concerns an industrial-strength case study in the emerging and challenging area of Real-time Online Interactive Applications (ROIA) which includes such popular and socially important applications as multi-player online computer games, high-performance simulations, e-learning, etc. There are several new challenges in these applications: thousands of simultaneous user connections to one application instance, optimized allocation of resources, and monitoring and steering to maintain the QoS parameters in dynamic distributed environments. The high-level scenario relates to four main stakeholders involved: (1) an *End-user* accesses ROIA sessions through graphical clients; (2) a *Scheduler* negotiates appropriate ROIA sessions based on the QoS requirements (e.g. connection latency, application domain, friends, expertise); (3) a *Hoster* (also called *provider*) is an organisation that provides a computational and network infrastructure for running ROIA servers; (4) a *Resource broker* provides a mechanism for application Schedulers and Hosters (and possibly other actors) to find each other in a large-scale environment and negotiate QoS relationships. These stakeholders accomplish a set up for an SBA, which is able to adapt during runtime, which aims for keeping a specified quality of service.

3.1.3.2 S-Cube Service Life Cycle Coverage

Authors in [16] apply the work given in [6] to the case study presented in [16] to addresses the issues related to the life-cycle. However, in the evaluation of the case study, the authors mainly focused on the adaptation rather than the evolution phase. The following table details the phases covered by the high-level scenario.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements	Identification of functional and non	End-users,	BPM

Engineering and Design	functional requirements relevant for the case study. Non-functional and adaptation requirements have been identified. The former are related to users (e.g., response time) and application, i.e., games (e.g., number of concurrent users, update rate). The latter refer to the way in which adaptation should be performed (e.g., need for proactive adaptation, level of autonomy)	Application Developer, SBA Engineer, Business Process Administrator	
Construction	On the basis of the identified requirements adaptation and monitoring mechanisms have been designed. In particular, to enable the service scalability to a high number and density of users, it is necessary to distribute game sessions adaptively, based on several adaptation strategies such as zoning [19] and replication [20]. A monitoring service observes the QoS parameters negotiated as performance contracts which must be preserved throughout the game session.	SBA Architect, Application Developer	SCC and SI
Deployment & Provisioning	All the designed mechanisms have been deployed as described in [16].	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	SCC
Operation & Management	During the *Operation and Management* phase, application is running and all the needed properties are monitored to identify conditions requiring adaptation or evolution.	End-users	SCC
Identify Adaptation Needs	As depicted in [16] we distinguish the following scenarios for triggering adaptation in ROIA applications: (i) Change in Quality of Service, (ii) Change in computational context, (iii) Change in business context, (iv) Prediction values from the capacity planning service. In details, we have defined specific adaptation trigger rules for each type of detected change.	-	SCC
Identify Adaptation Strategy	As suggested in [6] we linked the adaptation triggers with a set of suitable adaptation strategies.	-	SCC
Enact Adaptation	Supported by the impact analysis of each adaptation strategy.	-	SCC

Table 4: Detailed Coverage of the "Real-time Online Interactive Applications (ROIA)"-Scenario

3.1.3.3 Achieved Integration

In this high-level scenario, the authors demonstrate how the specific challenging features of ROIA can be met during the whole service life cycle. In particular, it was shown that adaptation mechanisms that help to develop high-quality, scalable ROIA applications can successfully be identified. Experimental results confirm the feasibility of integrating the work in accordance to the IRF life cycle model.

The scenario covers all phases, with an emphasis on the service composition and coordination layer (SCC).

3.1.4 High-Level Scenario: "Assumption-based Proactive Monitoring and Adaptation"

3.1.4.1 Detailed Scenario Description

Service-based systems often have to evolve dynamically and may have to change their behavior at runtime. This may happen due to various reasons, such as violation of service level agreements of the constituent services or unavailability of constituent services. Several efforts are found in the literature that deal with the dynamic adaptation of service based system [33][35][38][24]. However it should be appreciated that these issues sometimes may lead to unnecessary adaptation of the system, hence the adaptation should be planned and performed very carefully. For example, violation of service level agreements of a constituent service does not necessarily mean that the overall requirements of the service based system will be violated [33]. Again adaptation of a service based system by replacing an unavailable constituent service requires that the replacement service should be identified proactively to guarantee uninterrupted service from the service based system [36][39]. In this work we present an integrated framework that enables an application developer to design and implement service based system considering these adaptation triggering issues. A walkthrough of our approach along the S-Cube Service Life Cycle is given below:

In the evolution phase the application designer specifies a set of requirements for the service based system and a set of assumption under which the service based system should run. These requirements and assumptions are specified using template-based documents as introduced in [33]. The service based system is specified in a service composition language such as BPEL [41]. An initial SLA is negotiated and agreed for each constituent service used in the BPEL specification. SLA negotiation process takes into account the negotiation rules specified by the application designer and this process is carried out by exploiting the technique described in [36][42]. Once the service based system is designed it is formally verified to ensure that it satisfied the requirements under the specified assumptions. We apply the mechanism described in [43] for this formal verification. If the designed system passes the formal verification, the system is ready for deployment. Before the system is deployed a set of monitoring conditions is defined in order to monitor the system at runtime [33][40] and a service discovery query for each constituent service of the system is specified [36][42].

At runtime, in parallel with the execution of the service based system, a set of potential alternative services for each of the constituent service is identified and pre-agreed SLA is established with each identified service. Service discovery is driven by the discovery queries specified in the evolution phase [36][42]. Also at runtime of the service based system the assumptions are monitored [33][40]. If a violation of an assumption is detected, a runtime re-verification of the service based system is performed to check that the system still satisfies the overall requirements [33]. If the service based system fails to pass the re-verification, it is assumed an adaptation of the service based system is necessary. We apply the mechanisms described in [16] to identify the strategy that should be applied to adapt the service based system.

3.1.4.2 S-Cube Service Life Cycle Coverage

The following table details the phases covered by the high-level scenario.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Use of Quality Meta Model presented in [44] Use of template-based documents as introduced in	SBA Architect, Application Developer, SBA Engineer	BPM

	[33]		
Construction	Realization of workflow in BPEL [41] Specification of SLA negotiation rules and negotiate initial SLA [36][42] Specification of monitoring conditions [33][40].	SBA Architect, Application Developer	BPM SCC
Deployment & Provisioning	Check if workflow specification together with assumptions fulfill the SBA requirements [33] Specification of service discovery queries [36][42].	SBA Architect, SBA Modeler, Application Developer,	BPM SCC
Operation & Management	Monitoring during runtime [33][40] Identification of replacement services and establish pre-agreed SLA for each identified service [36][42].	-	SCC SI
Identify Adaptation Needs	Check if the workflow specification, the assumptions and the monitoring data fulfill the requirements [33]	-	BPM
Identify Adaptation Strategy	Service substitution as adaptation strategy [16]	-	SCC
Enact Adaptation	Use of proxy server supports replacement of services in service-based systems [38].	-	SCC SI

Table 5: Detailed Coverage of the "Assumption-based Proactive Monitoring and Adaptation"-Scenario

3.1.4.3 Achieved Integration

The current status of the integration reveals a good compatibility of the referenced research results. The monitoring component propagates monitoring information to the component which identifies adaptation needs. This component triggers the component which finds the adaptation strategy. Finally the adaptation is enacted. All artifacts which are necessary to achieve this, can successfully be produced in the foregoing phases during design time.

The high-level scenario covers all life cycle phases and functional layers. It shows a good coverage of the S-Cube live cycle.

3.1.5 High-Level Scenario: "QoS-driven multi-layer monitoring and adaptation"

3.1.5.1 Detailed Scenario Description

In this high-level scenario we focus on the QoS attributes of a Service Based System, as they emerge at the system's different layers: at the application layer and at the resource/infrastructure layer. The goal is to obtain a more precise estimate of the system's overall quality by reasoning at both of these layers. This scenario focuses on monitoring and adaptation, but also describes the artefacts needed to accomplish adaptation during the SBAs runtime. We monitor the quality of service attributes at both layers and devise special correlations between their behaviours. Then we adapt both layers accordingly, and in a coordinated fashion.

The scenario is based on techniques described in previous papers. In particular, we use the monitoring techniques described in [21] and [22]. We use correlation techniques as those described in [23]. We use techniques for discovering the adaptation needs, as seen in [24]. We use the software layer adaptation needs discussed in [25]. Finally, we use resource/infrastructure adaptation techniques such as [26] and [27].

3.1.5.2 S-Cube Service Life Cycle Coverage

The following table details the phases covered by the high-level scenario.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Use of the concepts expressed in [21] and [22] to define the monitoring requirements.	SBA Architect, Application Developer, Business Analyst, Business Process Analyst, Domain Expert, SBA Engineer	BPM, SCC and SI
Construction	Formalization using the semantics of [21] and [22]	SBA Architect, Application Developer	BPM, SCC and SI
Deployment & Provisioning	Use of concrete reference architectures from [21], [22], and [26]	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	BPM, SCC and SI
Operation & Management	Use of concrete reference architectures from [21] and [22] for runtime monitoring, and from [23] for correlation	SBA Architect, SBA Modeler, Service Provider, Service Architect	SCC and SI
Identify Adaptation Needs	Use of Process Factor Quality Analysis from [24]	SBA Architect, SBA Modeler, Service Architect	SCC and SI
Identify Adaptation Strategy	This is an important phase in our scenario. Currently no clear technique is already in place so we are looking at contributing with specific research effort to this step.	SBA Architect, SBA Modeler, Service Architect, Composition Designer	SCC and SI
Enact Adaptation	Use of software and resource/infrastructure adaptation from [25], [26], and [27]	SBA Architect, Service Architect, Service Provider, Composition Designer	SCC and SI

Table 6: Detailed Coverage of the "QoS-driven multi-layer monitoring and adaptation"-Scenario

3.1.5.3 Achieved Integration

The approach is still being evaluated and we are continuing to assess whether the approaches we have chosen can be integrated easily and effectively. We expect to produce one publication in one of the following conferences: International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2011), European Conference on Web Services (ECOWS 2011), International Conference on Service Oriented Computing (ICSOC 2011).

The high-level scenario covers all life cycle phases and functional layers. The scenario hence shows a good integration and coverage of the IRF life cycle.

3.1.6 High-Level Scenario: "Context-Aware Monitoring and Adaptation"

3.1.6.1 Detailed Scenario Description

The high-level scenario was derived composing the approaches proposed in [28] and [29]. The idea behind these approaches is that changes in the context of the SBA could require application adaptation and adaptation of the monitoring specifications. In particular, both the adaptations should take into account different types of context and changes at different layers during the application execution.
The starting point is the definition of a context model to capture all the main elements of the users and of the application, and the definition of patterns needed to put in relationships the changes in the context with the adaptation in the monitoring specification and the adaptation strategies.

The context model should capture all the dimensions that could trigger adaptation; six dimensions were identified to fulfill this need: Time, Ambient, User, Service, Business and Computational.
The monitor adaptation is obtained by the definition of some patterns for the adaptation of monitor rules; such rules are used by the monitor system. The adaptation of the monitor rules can be executed in three different ways, namely (a) by dynamic selection of the monitor rules to be used, (b) by automatic or semi-automatic modifications of monitor rules, and (c) by automatic or semi-automatic creation of new monitor rules.
Changes in the monitor rules are based on user interaction with a service-based system and different types of user context. The user context dimension includes role, skills, needs, preferences, and any other information needed to define the application context model. Moreover changes in some of the context properties could need the adaptation of the application. Different to this are the strategies and consequently the mechanisms to be enacted, depending on the context change occurred; some patterns should be defined in order to define such relationships.

3.1.6.2 S-Cube Service Life Cycle Coverage

The following table details the phases covered by the high-level scenario.

Service Life Cycle Phase	Argumentation that addresses the Life Cycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Definition of User-centric Context Model Definition of monitoring patterns Design of the context monitor and reasoner	SBA Architect, SBA Modeler, Application Developer, Business Analyst, Business Process Analyst, Domain Expert, SBA Engineer	SCC
Construction	Construction of contextual monitor and adaptation mechanisms	SBA Architect, Application Developer, Composition Designer	SCC
Deployment & Provisioning	Deployment of contextual monitoring and adaptation mechanisms	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	SCC
Operation & Management	Monitoring the application and the context at runtime	End User	SCC,SI
Identify Adaptation	Check if the application, the monitoring specifications, the	-	SCC, SI

Needs	monitoring data fulfill the requirements. Moreover check if the context changes require adaptation.		
Identify Adaptation Strategy	Choose a suitable adaptation strategy on the basis of the adaptation need.	End user (if needed)	SCC,SI
Enact Adaptation	Adaptation mechanisms to enact	-	SCC, SI

Table 7: Detailed Coverage of the "Context-Aware Monitoring and Adaptation"-Scenario

3.1.6.3 Achieved Integration

The high-level scenario derives from the mixing of the approaches presented in [28] and [29]. They seem to fit well together, since they address complementary issues related to the application adaptation based on context observation.

The presented high-level scenario covers all the phases of the S-Cube life cycle. Furthermore, all functional layers are addressed. The high-level scenario hence shows a good integration of the referenced results.

4 Conclusions and Recommendations

In this deliverable we present the rationalized validation approach, which instruments high-level scenarios based on the S-Cube life cycle. High-level scenarios relate to S-Cube research results and integrate them to one IRF life cycle walkthrough. The high-level scenarios, hence, indicate the integration of the research results. Beyond this, life cycle phases can be identified which are not well integrated.
The collected high-level scenarios are presented in section 3. The scenarios are dealing with the adaptation of service based applications during runtime. They describe the artifacts produced during design time needed to achieve runtime adaptation. In the following the high level scenarios and the achieved integrations are summarized briefly:

- The “Pro-active Adaptation based on the S-Cube Service Life Cycle”-scenario (cf. Section 3.1.1) enables proactive, assumption-based adaptation in order to permanently meet the SBA requirements. Every functional layer is addressed. The scenario shows a good integration of the research results and covers all phases of the life cycle. The scenario focuses on the service composition and coordination layer (SCC) and doesn’t cover the service infrastructure layer (SI).
- The “Design for Adaptation of Service-Based Applications”-scenario (cf. Section 3.1.2) proposes a holistic design method for adaptable service based applications, hence referring to research results, e.g. the ASTRO monitoring framework. This high-level scenario covers all phases of the IRF life cycle and functional layers. It demonstrates a good integration of the building blocks with targeting runtime adaptation of SBAs.
- The “Real-time Online Interactive Applications (ROIA)”-scenario (cf. Section 3.1.3) applies the design process described in the foregoing scenario (cf. Section 3.1.2) to the domain of ROIAs. It was shown that adaptation mechanisms that help to develop high-quality, scalable ROIA applications can successfully be identified. The scenario covers all phases, with an emphasis on the service composition and coordination layer (SCC).
- The “Assumption-based Proactive Monitoring and Adaptation”-scenario (cf. Section 3.1.4) reveals a good compatibility of the referenced research results produced within S-Cube. The high-level scenario covers all life cycle phases and functional layers. It addresses all life-cycle phases.
- The “QoS-driven multi-layer monitoring and adaptation”-scenario (cf. Section 3.1.5) is still being evaluated. The authors are continuing to assess whether the approaches can be integrated easily and effectively. The intermediate results, the high-level scenario is based on, cover all life cycle phases and functional layers. The scenario hence shows a good integration and coverage of the IRF life cycle.
- The “Context-Aware Monitoring and Adaptation”-scenario (cf. Section 3.1.6) mixes the approaches presented in [28] and [29]. The approaches seem to fit well together, since they address complementary issues related to the application adaptation based on context observation. The presented high-level scenario covers all the phases of the S-Cube life cycle. Furthermore, all functional layers are addressed. The high-level scenario hence shows a good integration of the referenced results.

These 6 high-level scenarios refer to a total of 19 S-Cube research results (including the three papers defining the High Level Scenarios). They integrate these results along the S-Cube Life-Cycle, facilitatating runtime adaptation. Three of these high-level scenarios already have been published. For the other three high-level scenarios publication plans exist.
The scenarios demonstrate a good integration of the S-Cube research results. Each presented high-level scenario covers both, the design and run time phase of the S-Cube IRF life cycle. All three functional layers (BPM, SCC and SI) are addressed. There is a slight focus on the service composition and coordination layer (SCC). This information will be provided to the joint research activities (JRAs) in order to clarify if any adjustments are necessary.

The collected S-Cube life-cycle stakeholder involvements are provided to JRA-1.1. Work package JRA-1.1 can use the stakeholders to advance the initial set of stakeholders.

References

[1] Mazza, V.; Pernici, B.: "Report on Common Pilot Cases". S-Cube Deliverable CD-IA-2.2.4, 2009

[2] Gehlert, A.; Pistore, M.; Plebani, P.; Versienti, L. (Eds): "First Version of the Integration Framework". S-Cube Deliverable CD-IA-3.1.3, 2009

[3] Basili, V.; Rombach, H.: The TAME Project: "Towards Improvement-oriented Software Environments". In: IEEE Transactions on Software Engineering, 14 (1988) 6, S.758–773.

[4] Heinrich, L.: "Bedeutung von Evaluation und Evaluationsforschung in der Wirtschaftsinformatik". In: Heinrich, L.; Häntschel, I. (Eds.): Evaluation und Evaluationsforschung in der Wirtschaftsinformatik: Handbuch für Praxis, Lehre und Forschung [Evaluation and Evaluation Research in Information Systems: Textbook for Industry, Teaching and Research]. Oldenburg, 2000, p. 7–22.

[5] Metzger, A.; Schmieders, E.; Cappiello, C.; Di Nitto, E.; Kazhamiakin, R.; Pernici, B.: "Towards Proactive Adaptation: A Journey along the S-Cube Service Life-Cycle". In: MESOA: 4th International Workshop on Maintenance and Evolution of Service-Oriented Systems. 2010

[6] Bucchiarone, A.; Cappiello, C.; Di Nitto, E.; Kazhamiakin, R.; Mazza, V.; Pistore, M.: "Design for Adaptation of Service-Based Applications: Main Issues and Requirements". ICSOC/ServiceWave Workshops 2009: 467-476.

[7] Mittal, K.: Service oriented unified process, http://www.kunalmittal.com/html/soup.html.

[8] Trainotti, M.; Pistore, M.; Calabrese, G.; Zacco, G.; Lucchese, G.; Barbon, F.; Bertoli, P.; Traverso, P.: "ASTRO: Supporting Composition and Execution of Web Services". In: Benatallah, B.; Casati, F.; Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 495–501. Springer, Heidelberg (2005)

[9] Linner, D.; Pfeffer, H.; Radusch, I.; Steglich, S.: "Biology as Inspiration: Towards a Novel Service Life-Cycle". In: Xiao, B.; Yang, L.T.; Ma, J.; Muller-Schloer, C.; Hua, Y. (eds.) ATC 2007. LNCS, vol. 4610, pp. 94–102. Springer, Heidelberg (2007)

[10] Spanoudakis, G.; Zisman, A.; Kozlenkov, A.: "A Service Discovery Framework for Service Centric Systems". In: IEEE International Conference on Services Computing, vol. 1, pp. 251–259 (2005)

[11] Verma, K.; Gomadam, K.; Sheth, A.P.; Miller, J.A.; Wu, Z.: "The METEOR-S Approach for Configuring and Executing Dynamic Web Processes". Technical report, University of Georgia, Athens (June 2005).

[12] Baresi, L.; Guinea, S.; Pasquale, L.: "Self-healing BPEL processes with Dynamo and the JBoss rule engine". In: ESSPE 2007, pp. 11–20. ACM, New York (2007)

[13] Colombo, M.; Di Nitto, E.; Mauri, M.: Scene: "A service composition execution environment supporting dynamic changes disciplined through rules". In: Dan, A.; Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 191–202. Springer, Heidelberg (2006)

[14] Erradi, A.; Maheshwari, P.; Tosic, V.: "Policy-driven middleware for self-adaptation of web services compositions". In: van Steen, M.; Henning, M. (eds.) Middleware 2006. LNCS, vol. 4290, pp. 62–80. Springer, Heidelberg (2006)

[15] Rukzio, E.; Siorpaes, S.; Falke, O.; Hussmann, H.: "Policy based adaptive services for mobile commerce". (2005)

[16] Meiländer, D.; Gorlatch, S.; Cappiello, C.; Mazza, V.; Kazhamiakin, R.; Bucchiarone, A.: “Using a Lifecycle Model for Developing and Running Highly Interctive Distributed Applications”. In: Proceedings of ServiceWave 2010.

[17] The edutain@grid project, http://www.edutaingrid.eu, 2009.

[18] Gorlatch, S.; Glinka, F.; Ploss, A.; Müller-Iden, J.; Prodan, R.; Nae, V.; Fahringer, T.: “Enhancing Grids for Massively Multiplayer Online Computer Games.” In: Euro-Par 2008 - Parallel Processing, volume 5168 of Lecture Notes in Computer Science, 2008.

[19] Cai, W.; Xavier, P.; Turner, S. J.; Lee, B.-S.: “A scalable architecture for supporting interactive games on the internet”. In: 16th Workshop on Parallel and Distributed Simulation, pages 60-67, Washington, DC, USA, 2002. IEEE Computer Society.

[20] Müller-Iden, J.; Gorlatch, S.: “Rokkatan - Scaling an RTS game design to the massively multiplayer realm”. In: Computers in Entertainment, 4(3):11, 2006.

[21] Baresi, G.; Pistore, M.; Trainotti, M.: “Dynamo + astro: An integrated approach for bpel monitoring”. Web Services, IEEE International Conference on, vol. 0, pp. 230–237, 2009.

[22] Baresi, G.; Kazhamiakin, R.; Pistore, M.: “An integrated approach for the run-time monitoring of bpel orchestrations”. In: Service-Wave ’08: Proceedings of the 1st European Conference on Towards a Service-Based Internet. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 1–12.

[23] Baresi, G.; Caporuscio, M.; Ghezzi, C.; Guinea, S.: “Model-driven Management of Services”. Proc. 8th European Conference on Web Services (ECOWS), 2010.

[24] Kazhamiakin, R.; Wetzstein, B.; Karastoyanova, D.; Pistore, M.; Leymann, F.: “Adaptation of service-based applications based on process quality factor analysis”. In: Proc. Of 2nd Intl. workshop on Monitoring, Adaptation and Beyond (MONA+), Collocated with ICSOC/Service-Wave’09, 2009

[25] Bertoli, P.; Kazhamiakin, R.; Paolucci, M.; Pistore, M.; Raik, H.; Wagner, M.: “Control Flow Requirements for Automated Service Composition”. ICWS 2009

[26] Kertesz, A.; Kecskemeti, G.; Brandic. I.: “Autonomic SLA-aware Service Virtualization for Distributed Systems”. Euromicro International Conference on Parallel, Distributed and Network-Based Computing

[27] Gauvrit, G.; Daubert, E.; André. F.: “SAFDIS: A Framework to Bring Self-Adaptability to Service-Based Distributed Applications”. Euromicro Conference on Software Engineering and Advanced Applications 2010

[28] Contreras, R.; Zisman, A.: “A Patterns-based Approach for Monitoring Adaptation”. SwSTE10.

[29] Bucchiarone, A.; Cappiello, C.; Di Nitto, E.; Kazhamiakin R.; Mazza, V.: “A context-driven Adaptation Process for Service Based Application”. PESOS 2010

[30] Ejarque, J.; Micsik, A.; Sirvent, R.; Pallinger, P.; Kovacs, L.; Badia, R. M.: “Semantic Resource Allocation with Historical Data Based Predictions”. The First International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2010 November 21-26, Lisbon, Portugal

[31] Leitner, P.; Wetzstein, B.; Rosenberg, F.; Michlmayr, A.; Dustdar S.; Leymann, F.: “Runtime prediction of service level agreement violations for composite services”. In: 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing, 2009

[32] Zeng, L.; Lingenfelder, C.; Lei, H.; Chang, H.: “Event driven quality of service prediction”. In: ICSOC 2008, ser. Lecture Notes in Computer Science, A. Bouguettaya, I. Kruger, and T. Margaria, Eds., vol. 5364, 2008, pp. 147–161.

[33] Gehlert, A.; Bucchiarone, A.; Kazhamiakin, R.; Metzger, A.; Pistore, M.; Pohl, K.: "Exploiting assumption-based verification for the adaptation of service-based applications". In: SAC '10: Proceedings of the 2010 ACM Symposium on Applied

[34] Oriol, M.; Franch, A.: "Monitoring Adaptable SOA-Systems using SALMon". (MONA+'08).

[35] Burgstaller et al.: "Monitoring and Adaptation of Service-oriented Systems with Goal and Variability Models". UPC TR-09-8

[36] Mahbub, K.; Spanoudakis, G.: "Proactive SLA Negotiation for Service Based Systems". SERVICES 2010.

[37] Gehlert, A.; Metzger, A.: "Quality reference model for SBA". Deliverable CD-JRA-1.3.2, 2008. [Online]. Available: http://www.scube-network.eu/results/

[38] Mahbub A.; Zisman, A.: "Replacement Policies for Service-Based Systems". In: 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), co- located with ICSOC 2009, 2009.

[39] Zisman, A.; Spanoudakis, G.; Dooley, J.: "A Framework for Dynamic Service Discovery". 23rd Int. IEEE/ACM Conf. on Automated Software Engineering, 2008

[40] Oriol, M.; Franch, X.; Marco, J.: "SALMon: A SOA System for Monitoring Service Level Agreements". LSI-10-18-R, 2010

[41] OASIS Standard: Web services business process execution language version 2.0, 11 April 2007.

[42] Mahbub, K.; Spanoudakis, G.: "A Framework for Proactive SLA Negotiation". 5th International Conference on Software and Data Technologies (ICSOFT 2010), 22 - 24 July, Athens, Greece

[43] Kazhamiakin, R.; Pistore, M.: "Static verification of control and data in web service compositions". In: ICWS '06: Proceedings of the IEEE International Conference on Web Services, pages 83-90, Washington, DC, USA, 2006. IEEE Computer Society.

[44] Gehlert, A.; Metzger, A.: "Quality reference model for SBA". Deliverable CD-JRA-1.3.2, 2008. [Online]. Available: http://www.scube-network.eu/results/

[45] Bianculli, D.; Ghezzi, C.; Spoletini, P.; Baresi, L.; Guinea, S.: "A Guided Tour through SAVVYWS: a Methodology for Specifying and Validating Web Service Compositions". In: Advances in Software Engineering, ser. Lecture Notes in Computer Science. Springer-Verlag, 2008, vol. 5316, pp. 131–160.

[46] Baresi, L.; Guinea, S.; Pasquale, L.: "Integrated and composable supervision of bpel processes". In: International Conference on Service-Oriented Computing (ICSOC), 2008, pp. 614–619.

[47] Karastoyanova, D.; Leymann, F.: "Bpel'n'aspects: Adapting service orchestration logic". In: International Conference on Web Services (ICWS), 2009, pp. 222–229.

[48] Ghezzi C.; Guinea, S.: "Run-time monitoring in service-oriented architectures". In: Test and Analysis of Web Services, L. Baresi and E. D. Nitto, Eds., 2007, pp. 237–264.

[49] Barbon, F.; Traverso, P.; Pistore, M.; Trainotti, M.: "Run-Time Monitoring of Instances and Classes of Web Service Compositions". In: IEEE International Conference on Web Services (ICWS 2006), 2006, pp. 63–71.

[50] Metzger, A.; Sammodi, O.; Pohl, K.; Rzepka, M.: "Towards proactive adaptation with confidence augmenting service monitoring with online testing". In: Proceedings of the ICSE 2010 Workshop on Software Engineering for Adaptive and Self-managing Systems (SEAMS '10), Cape Town, South Africa, 2-8 May 2010.

[51] Hielscher, J.; Kazhamiakin, R.; Metzger, A.; Pistore, M.: "A framework for proactive self-adaptation of service-based applications based on online testing". In: ServiceWave 2008, ser. LNCS, no. 5377. Springer, 10-13 December 2008.

[52] S-Cube Partners: “Initial Definition of User Patterns”. Contractual Deliverable PO-IA-3.1.4, S-Cube Network of Excellence, June 2010.

[53] S-Cube Partners: “Separate Design Knowledge Models for Software Engineering & Service-Based Computing”. Contractual Deliverable CD-JRA-1.1.2, S-Cube Network of Excellence, December 2008.

[54] Qing Gu and Patricia Lago: “A Stakeholder-Driven Service Life Cycle Model for SOA”. In Proceedings of the 2nd International Workshop on Service-Oriented Software Engineering (IW-SOSWE’07), pages 1–7, Dubrovnik, Croatia, 2007.

[55] Raman Kazhamiakin (Eds): “Results of the first Validation”. S-Cube Deliverable CD-IA-3.2.2, 2010

[56] S-Cube Partners: “Assessment report and adjustment action plan”. S-Cube Deliverable CD-Mgt-1.3.1, 2010

[57] Raman Kazhamiakin (Eds): “Comprehensive, integrated adaptation and monitoring principles, techniques and methodologies across functional SBA layers considering context and HCI”. S-Cube Deliverable CD-JRA-1.2.5