Grant Agreement N° 215483

Title:	*Report on the Final Validation of the Integration Framework and Empirical Evaluation of the Integration Framework*
Authors:	*UniDue, POLIMI, CITY, FBK, CNR, UPM, Tilburg, Lero, UoC, INRIA, TUW, SZTAKI, USTUTT*
Editors:	*Elisabetta Di Nitto (POLIMI), Eric Schmieders (UniDue)*
Reviewers:	*Antonio Bucchiarone (FBK), Damian A. Tamburri (VUA)*
Identifier:	*CD-IA-3.2.5*
Type:	*Deliverable*
Version:	*1*
Date:	*14 March 2012*
Status:	*Final*
Class:	*External*

Management Summary

In this document, we report about the validation activities performed in the 4th project year as part of WP-IA-3.2. The deliverable describes the vision and strategy of the work package (using the Description of Work Amendment #4 as basis), provides a description of the work package roadmap until the end of the S-Cube project introduces and describes the application of the validation, and, finally, summarizes results and provides some conclusions.

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 (Empirical validation of the research results): The building blocks of the S-Cube Integration Research Framework (IRF) are *validated empirically*. 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 (Validation of the integration of the building blocks in the integration framework): The assembly of the building blocks in the IRF is validated through suitable high-level scenarios analysed along the service lifecycle. The definition of the validation scenarios starts with a collection and analysis of existing scenarios through a systematic survey. Next existing scenarios are analysed and extended, or in case appropriate ones are lacking, new scenarios are devised. The ultimate goal of the validation is to revise and improve the IRF. For this reason it will be conducted iteratively in different phases through the duration of the activity. Furthermore, the task comprises the collection of stakeholders associated to the high-level scenarios.

Together, these tasks contribute to the consolidation of the IRF (IA-3.1). In the following they will be described in greater detail.

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

The aim of this task is the evaluation of the building blocks of the IRF. A specific empirical evaluation may focus on a single building block or may cross-cut multiple building blocks. Methods for the empirical evaluation 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 and to provide a structured access to validation results (e.g., by linking from research questions in the IRF to papers that include the validations).
This task is composed by the following activities:

- Set up empirical evaluations of (parts of) the research results within the IRF.
- Support the organisation, implementation and execution of the evaluation activities.
- Analyse the results of the evaluation activities.
- Provide a structured assessment of 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 whole Integrated Research Framework (IRF). To this end, high-level scenarios will be employed to check that the relevant activities of the service lifecycle are covered. Those high-level scenarios will be defined in close cooperation with WP-JRA-1.1.
The task iteratively takes place during the different phases of the network. In each iteration, the task is organized in the following steps:

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

[1] Deliverable CD-IA-3.2.5 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.

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

This work package offers to the S-Cube convergence knowledge model the set of obtained scenarios.

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 is 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 anymore relevant 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 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. 0).
- *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 has been used for WP-IA-3.2 for the years 3 and 4 (cf. Figure 1).

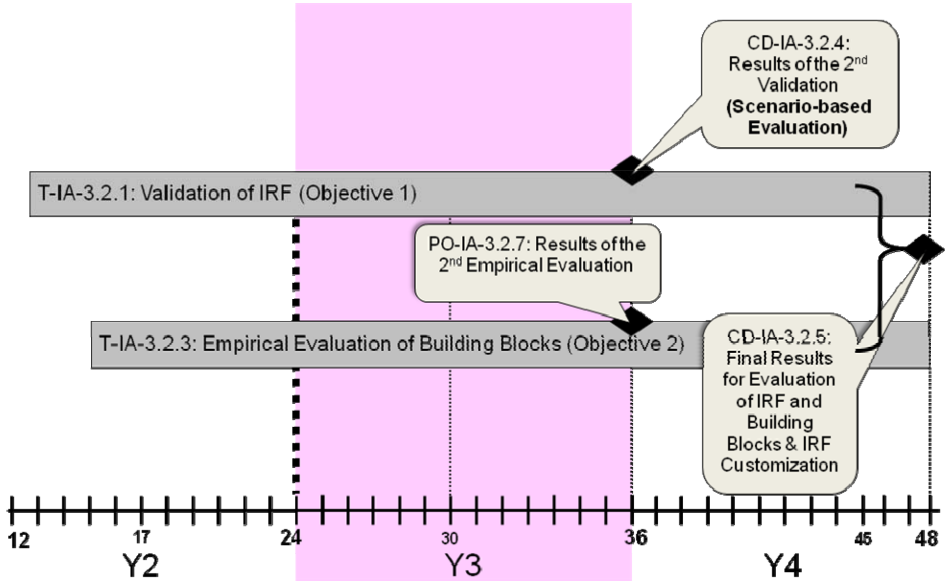


Figure 1: Timeline of IA-3.2

In the final year (Y4) the high-level scenario-driven validation is applied again. The current deliverable reports on the results of such validation.

1.6 Document Structure

This document is structured as follows:

- Section 2 presents an overview of the approach we have adopted for validating each IRF building blocks as well as their integration.
- Section 3 describes all validation results that we have achieved. For the sake of homogeneity, each result is described through a precise format. All results are also summarized in two synoptic tables.
- Finally, Section **Fehler! Verweisquelle konnte nicht gefunden werden.** provides recommendations and conclusions.

2 Validation Strategies

The validation strategy for IA-3.2 was introduced in CD-IA-3.2.2 [1] 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 [2]). Together with the refocusing 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 Strategy for the empirical validation of the IRF building blocks

In order to validate the IRF building blocks, the evaluation results need to be documented in a uniform way. For this documentation we use the structural template depicted in Table 1. This allows evaluation participants to precisely document the setup of the empirical evaluation, to analyze the results of the evaluation activities, and to provide a structured assessment of the evaluation results.

Table 1. Structure for Documenting Validation Results.

Validation Set-up & Result	
Name	Name of the validation result.
Synopsis	Brief summary of the validation result (1 – 2 sentences).
Authors	List of authors of the validation result.
Research questions	Reference to research questions specified in the IRF related to the validation result.
Case Study	Used case study. Please specify if the scenario is new or listed here: http://scube-casestudies.ws.dei.polimi.it/index.php/Main_Page
Scenario	Scenario used in the evaluation and any related information (e.g. how the scenario is used in the evaluation exercise).
Method	The empirical technique used for validation, i.e. one of: experiment, case study, field study, prototype, proof (see below).
Description	Short description of the validation result.
Goal	Description of the goal and objectives associated to the validation.
Set-up	Description of the set-up of the validation (e. g. settings - lab, organizations departments; the tools used - computational platform, technical specifications if appropriate etc.)
Inputs	Description of the materials (e. g., data) used in the validation.
Outputs	Description of the outputs (e. g., results) of the validation.
Outcome	One of: positive (the research result fulfills its goal), negative (the research result does not fulfill its goal).
Experiences	Comments and experiences on the validation (both positive and negative) gained performing the validation and that may inform a replication of the exercise
References	List of link to the paper and/or web page, in which this result was used.
Glossary	References to relevant terms in the knowledge model.
Keywords	List of keywords to facilitate search.

Since the terminology for the different research methods may differ in the various S-Cube research communities we use the following definitions in WP-IA-3.2:

- *Controlled Experiment*: a controlled experiment is a research method carried out in a laboratory environment. It aims to test a hypothesis by manipulating its independent variables and measuring its dependent variables.
- *Case Study:* in a case study a single phenomenon is studied in a real-life context (e.g. in an organization). The researcher doing a case study only observes the real-life context.

- *Field Study*: a field study is the broader version of a case study where multiple phenomena are studied in different real-life contexts (e.g. in different organizations). The researcher doing a field study only observes the real-life context.
- *Action Research:* in action research the researcher attempts to solve a real-world problem while simultaneously studying the experiences gained during the solving process. The researcher doing action research actively participates in the problem solving process.
- *Survey:* a survey uses a structured questionnaire to capture data from different individuals, e.g. by sending the questionnaire via mail to organizations or by using a web questionnaire.
- *Literature review:* a literature review analyses the state of the art in a certain field obtaining as a result a classification of the exiting approaches and a deeper understanding of the field.
- *Prototyping:* prototyping is used to realize some aspects of an envisioned system (or algorithm) in a demonstrator or prototype to show the feasibility of the approach.
- *Experiments with Prototypes:* existing prototypes may be used to carry out experiments demonstrating the superiority of an algorithm or system—especially in cases where a formal proof is not feasible.
- *Formal Proof:* a formal proof is a mathematical method to formally demonstrate that a (formal) system fulfills certain properties.
- *Proof of concept:* a proof of concept is evidence which demonstrates that the concept being proposed (e.g. an approach, an algorithm) is feasible and viable.

The above classification will be used in Section 3.1 to cluster the validation experiments that have been performed in the fourth year of the project. Of course, given the large variety of possible experiments, not all fields in the table are mandatory, in particular, the field concerning the experiences is filled only when relevant.

2.2 Strategy for validating the assembly of the IRF building blocks

2.2.1 Use of High-Level Scenarios along the S-Cube Lifecycle

In this deliverable we use the S-Cube lifecycle 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 lifecycle view is ideal to understand, align, integrate and discuss the research outcomes. The service lifecycle model envisioned by the S-Cube framework is shown in Figure 2. It captures an iterative and continuous method for developing, implementing, and maintaining service-based applications 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 lifecycle, please consult PO-IA-3.1.4 [3].

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) lifecycle phases. This integration we consider as high-level scenario.

Each high-level scenario refers to several results produced within S-Cube and aligned to the related phase of the lifecycle. The results addressed in the high-level scenarios are paper based, i.e. determined from publications by members of the S-Cube network.

2.2.1.1 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 lifecycle.

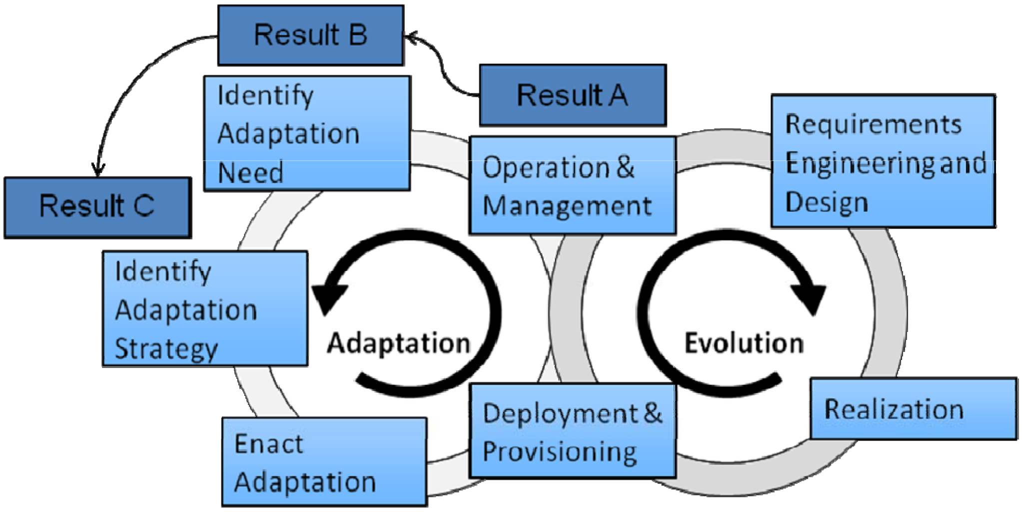


Figure 2: High-Level Scenarios

This high-level scenario reveals an integration of the three addressed phases. In consequence, once the high-level scenario is structured along the lifecycle, we can identify lifecycle 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 (Business Process Management – BPM, Service Composition and Coordination – SCC, Service Infrastructure - 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.

2.2.1.2 Methodology

We collect the high-level scenarios by using a template. The template comprises three sections: “Detailed Scenario Description”, “S-Cube Lifecycle Coverage” and “Achieved Integration”. The template allows structuring the results along the lifecycle.

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, named “S-Cube Lifecycle Coverage”,

Table 2 is used to structure the description of the high-level scenario along the lifecycle declaring the related S-Cube functional layers (BPM, SCC and SI). The table also requires information about the functional layers affected by the scenario and the stakeholders having a relevant role in the scenario. These are selected from a list defined in PO-IA-3.1.4 [3] and summarized in the following section.
Finally, in the “Achieved integration” section, the achieved integration and the open issues for each high-level Scenario are summarized.

Table 2. Template for structuring the research along the S-Cube lifecycle.

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

The high level scenarios are presented in Section 3.2.

2.2.2 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 2.2.1). To achieve this, we provide the stakeholder types defined in PO-IA-3.1.4 [3] (cf. Figure 3) to the authors.

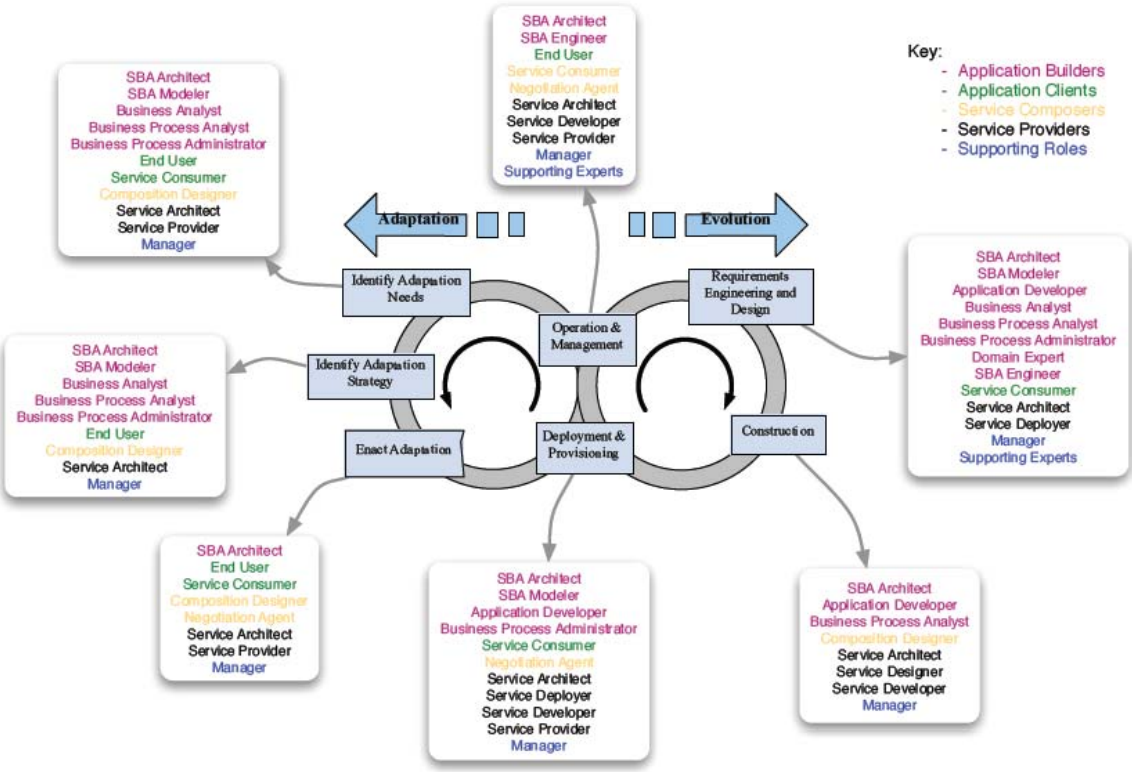


Figure 3: S-Cube Lifecycle Phases with Stakeholder Types

2.3 Collection process

This deliverable reports about two types of validation experiments, those concerning specific elements of the S-Cube Integration Framework and those focusing on the framework as a whole. Both types of validation experiments are performed as part of the normal activities conducted in the S-Cube research work packages. IA-3.2 observes the activities of the research work packages and collects information about validation experiments by periodically requesting all S-Cube participants to fill in the templates that have been presented in Section 2.1 and 2.2. For the purpose of this specific deliverable the collection process has been started in October 2011 and has ended at the beginning of February 2012.

3 Validation Results

This section reports about the two validation activities carried out in the 4th project year as part of WP-IA-3.2: those concerning specific elements of the S-Cube Integration Framework (Section 3.1) and those focusing on the framework as a whole (Section 3.2).

3.1 Empirical validation of the IRF building blocks

In this section all experiments are reported based on the validation method they have adopted. Each experiment is described using the tabular structure described in Section 2.1.

3.1.1 Validation Method: Controlled Experiment

3.1.1.1 Mining Lifecycle Event Logs for Enhancing Service-based Applications

Validation Set-up & Result	
Name	Mining Lifecycle Event Logs for Enhancing Service-based Applications.
Synopsis	The approach aims at extracting useful knowledge (i.e., *patterns*) from the historical usage data collected by SBA event logs, thus to help SBA designers with novel service recommendation tools.
Authors	Leitner, P., Nardini, F.M., Tolomei, G., Dustdar, S., Silvestri, F.
Research questions	Process Mining
Case study	Not Present.
Scenario	We use the VRESCo system. VRESCo is an experimental runtime environment developed at Vienna University of Technology under an open source license. The project aims at solving some of the research problems identified in (Papazoglou, 2007), e.g., dynamic selection of services based on Quality-of-Service (QoS), dynamic rebinding and service composition, service metadata and event-based services computing. Our scenario refers to event-based service-oriented architectures. In the VRESCo system, events of various types are triggered.
Method	Experiment.
Description	We treat the process mining problem as an instance of the sequential pattern mining problem. Therefore, we discover sequences of frequently-invoked services by applying two well-known algorithms to a real-world dataset, i.e., PrefixSpan and MiSTA to logs coming from a VRESCo runtime environment.
Goal	Apply data mining algorithms and techniques to a real-world service event log. Analyze it in order to discover software services that are frequently invoked and composed together.
Set-up	Validation of our hypothesis has been conducted on a real-world dataset from the VRESCo infrastructure, which has been previously adapted for being processed by mining algorithms. Experiments have been run on a Linux Ubuntu 10.04.3/x86 64 bit platform.
Inputs	VRESCo event log dataset.
Outputs	Set of patterns describing frequent sequences of invoked services.
Outcome	Positive.
Experiences	We are confident that the results we obtained could be helpful at design-time, when SBA developers could be provided with novel service recommendation tools that are able to predict and suggest next services that should be included in the current service workflow composition. This work might be extended in order to discover sequences of service

	invocations that lead to failures or unexpected behaviors. This knowledge could be exploited either for preventing SBA designers to deploy possible faulty service compositions as well as for devising novel run-time adaptation mechanisms in response to undesired events.
References	Leitner, P., Nardini, F.M., Tolomei, G., Dustdar, S., Silvestri, F.*: Mining Lifecycle Event Logs for Enhancing Service-based Applications*. In WAS4FI 2011. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: *Service-Oriented Computing: State of the Art and Research Challenges*. IEEE Computer 40(11), 38–45 (November 2007)
Glossary	Process mining.
Keywords	Service event log mining, frequent service sequence, sequential pattern mining.

3.1.1.2 Resource management with license reservation

Validation Set-up & Result	
Name	Resource management with license reservation
Synopsis	A semantic resource allocation approach is extended to support the scheduling of software license resources. This approach defines the licenses as another type of computational resource which is available in the system and must be allocated to the different service requests. The license scheduling has been implemented using SPARQL queries and RDF rules, which evaluate the semantic license terms and check the availability of licenses.
Authors	Jorge Ejarque, Andras Micsik, Raul Sirvent, Peter Pallinger, Laszlo Kovacs and Rosa M. Badia
Research questions	Service composition run-time validation of non-functional requirements Supporting adaptation of service-based applications Runtime SLA Violation Prevention
Case Study	New: allocating resources in a job execution system.
Scenario	Computational tasks are submitted to the system with specified need for licensed software, and other SLAs (e.g. deadline). System has a predefined set of available licenses with specific license usage restrictions. The approach allocates the licenses to tasks and monitors timely execution of tasks, if needed adapts the license allocation or task execution.
Method	Experiment
Description	The tool successfully computed the license allocations, and avoided the violation of SLAs. The overhead of reasoning and semantic modelling applied was acceptible.
Goal	Check the feasibility of the approach and the expressivity of semantic license model.
Set-up	Tools used: JADE multi-agent platform, Jena Semantic Web framework, existing test environment with 6 hosts. Location: UPC.
Inputs	Semantic software license descriptions, semantic task descriptions including agreed SLA, semantic host descriptions.
Outputs	Executed schedules of tasks including host, licenses used and timestamps.
Outcome	positive
Experiences	N/A
References	Jorge Ejarque, András Micsik, Raúl Sirvent, Peter Pallinger, László Kovács, Rosa M. Badia: *Job Scheduling with License Reservation: A Semantic Approach*. PDP 2011: 47-54
Glossary	Self-Adaptation, Proactive Adaptation, Semantic Web, Adaptability, Service Level Agreement (SLA), SLA violation, Service-Based Application (SBA)
Keywords	multi-agent, semantics, scheduling, resource allocation, software licenses, grid computing, cloud computing, distributed systems

3.1.2 Validation Method: Case Study

3.1.2.1 Socio-technical dynamics of a public service through actor-network theory

Service network environments remain one of the most significant, yet 'invisible' infrastructures within the modern business era. However, although the emerging paradigm of 'service science' calls for more theoretical focus on understanding complex service systems, few efforts have surfaced which apply a new theoretical lens on understanding the underlying trajectories of socio-technical dynamics within a service system. This empirical research examines the socio-technical impact of technology on public service network dynamics. In particular, it explores an academic service network, paying specific attention towards a critical end-to-end exam grading process. The research adopts actor-network theory (ANT) as a research lens, which offers a rich vocabulary to describe the interplay of socio-technical dynamics, which influence the service system reconfiguration. In addition, the research offers a visual comparison of the service relational infrastructure through the use of social network analysis (SNA), which complements ANT to examine a pre- and post-technological implementation.

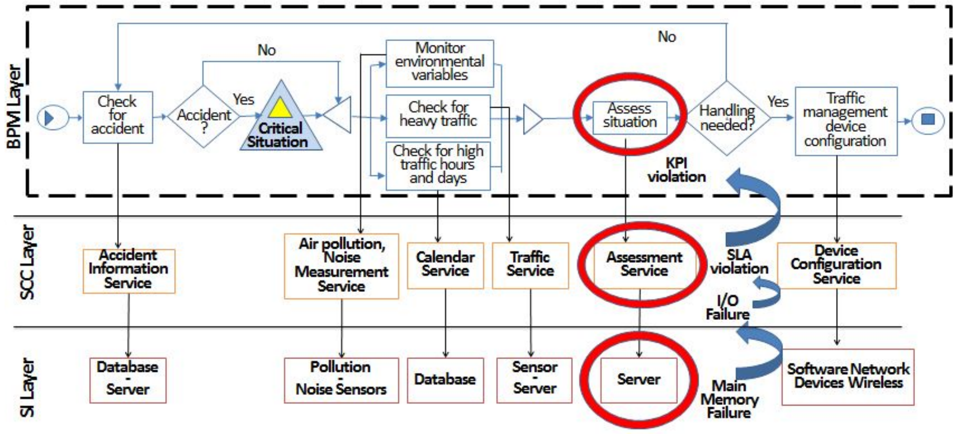
Validation Set-up & Result	
Name	Socio-technical dynamics of a public service through actor-network theory
Synopsis	The research presents a theory which examines the translation process of IT-enabled public service network innovation. In essence, the research explains how public service technological innovations commands control over public sector behavior and therefore acts as an agent of bureaucracy which alters the relational dynamics of power, risk, responsibility, and accountability.
Authors	Noel Carroll
Research questions	How do socio-technical dynamics impact the service network translation process for implementing IT innovation?
Case Study	The case study examines an academic service network. This empirical research explores an academic service network, with particular attention paid towards a critical end-to end exam grading process. The researcher employs a single case study to examine the impact of a Web-based system on a traditionally bureaucratic public service system and its transformation from a paper-based system to an automated system.
Scenario	I was presented with a unique opportunity to model the grading service network within an academic environment to explore the change and reformation of an actor-network through IT-enabled innovation.
Method	Case study
Description	This research examines the impact of implementing IT on service network dynamics. Interestingly, the literature indicates that in recent years the distinction between business services and IT services is becoming less apparent, and this generates some overlap. This is particularly true with the complexity of service networks. Networks are created when actors interact and often the key to understanding networks is when results are analyzed to derive facts about the network. For example, we can explain and model a structural analysis to identify the significant positions of an actor-network. In addition, if we know facts about the network actors or the nature of interaction, we can derive other facts or examine other phenomena. Networks are formed through the collaboration of people to deliver a service. Nowadays, organizations are exploring methods to introduce more innovative techniques to deliver services (for example, efficiency and value-added factors) within dynamic networks. Many organizations are opting for IT-enabled networks to foster innovation within a service network.

	However, one of the critical questions is: *How does the introduction of IT impact on service relationships in the service network?* Thus, the contribution of this research is to presents different insights from the traditional service management literature (strategy and technology) since it pays equal attention to both human and non-human entities (i.e. actors) which configure and form what we describe as a 'service'.
Goal	• Examine the formation of a public service network and the impact of IS on the network through the material semiotic perspective. • Identify the main socio-technical factors which contribute towards service network assemblages within the public sector. • Determine how relational ties between actors within a service network contribute towards service socio-technical dynamics. • Examine how IS impacts the relational ties between service actors within a service network. • Develop theory on socio-technical service network assemblages within the public sector.
Set-up	Case study: 60 Interviews (30 pre- and 30 post IS implementation), 200 online surveys respondents, observation, secondary data, SNA mapping, NVivo qualitative data analysis. Followed guidelines and techniques explained by Miles and Huberman (1984).
Inputs	Empirical data gathered for qualitative analysis as listed in set-up.
Outputs	The research presents a theory on the translation process of IT-enabled public service network innovation. In essence, it explains how public service technological innovations commands control over public sector behavior and therefore acts as an agent of bureaucracy which alters the relational dynamics of power, risk, responsibility, and accountability. The research also provides a visual comparison of the impact of IT through service innovation using SNA. Through the use of SNA, I have introduced service network relational analytics which reports on the change in service interaction as a result of IT-enabled service innovation.
Outcome	Positive
Experiences	I feel that the research offers a unique approach towards modeling and explaining service network formation while it considers the socio-technical factors of service dynamics.
References	Papers submitted for review
Glossary	Business process management, service network analytics, Actor-network theory, social network analysis, service science, service relational structure, public service, case study.
Keywords	Actor-network theory, social network analysis, service science, service relational structure, public service, case study.

3.1.2.2 Event-Based Cross-Layer Service Monitoring and Adaptation

Cross-layer monitoring and adaptation has gathered a great deal of interest in SOA research. This approach proposes a framework, able to monitor and adapt SBAs across all functional layers, by using techniques such as event monitoring and logging, event-pattern detection, and mapping between event patterns and appropriate adaptation strategies.

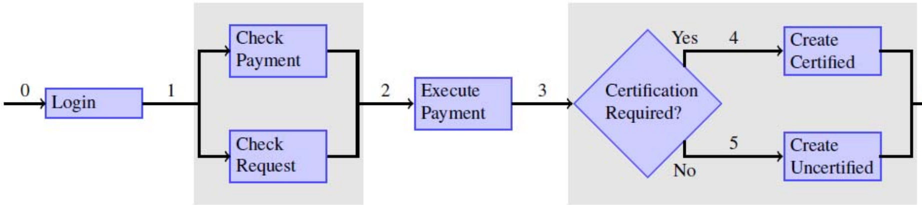
Validation Set-up & Result	
Name	Event-Based Cross-Layer Service Monitoring and Adaptation
Synopsis	The approach proposes a cross-layer monitoring and adaptation framework, which is validated by a case study.
Authors	Zeginis Chrysostomos, Konsolaki Konstantina, Kritikos Kyriakos, Plexousakis Dimitris
Research	Cross-layer SBA adaptation

questions	Predictive SBA monitoring techniques Cross-layer integrated monitoring mechanisms Supporting adaptation of service-based applications
Case study	Traffic Management Case study: http://scube-casestudies.ws.dei.polimi.it/index.php/Traffic_Case_Study
Scenario	We elaborate a cross-layer scenario on Traffic Management domain. The scenario model is depicted in the figure below.  In this example the boxes denote: • Specific activities in the Business Process at the BPM layer • Concrete services that implement the corresponding activities, at the SCC layer • The underlying infrastructure at the SI layer Suppose that a KPI exists dictating that the maximum duration of the process should be less than 10 seconds. Further, suppose that an SLA exists for the assessment service (AS) dictating that its maximum execution time must be less than 6 seconds. Thus, as it can be seen, a violation of the respective SLA constraint may cause a violation to the KPI, by considering that the previous process activities do not run longer than 3 seconds. It must also be indicated that AS's execution time is inverse proportional to the main memory size and the CPU percentage allocated for its execution. Moreover, there is a low limit for the main memory allocated, after which the SLA violation will be unavoidable as the service behavior will be unpredictable and even if it does not fail it will certainly take a long time to execute. In fact, after 2 seconds from the AS's execution, the main memory allocated to it has indeed surpassed the low level of 50 MB. A Monitoring Component, running at the server where AS is executed, detects that the available main memory is not sufficient (SI layer) for AS. At the same time, another Monitoring Component detects that there is an I/O failure at the SCC layer as AS has produced a wrong output. Both events are first sent to the Translator, which transforms them to the appropriate format and sends them to the Reasoner. Based on the two events received, a specific rule is fired which derives that the best strategy is to execute another instance of the AS service at a more powerful server and with a better memory and CPU allocation. The suitability of the strategy lies on the fact that by executing a "better" service instance and with better allocation for the hardware resources, the probability that the SLA is not violated becomes very high (as we do not know if another failure may occur in the near future regarding the new instance) and in this way also the KPI violation may be avoided. Such a rule has been derived by the Event Pattern Detector based on the previous history log and has been already inserted into the Reasoner. The derived strategy is sent to the Adaptation Manager which executes it with the assistance of the Infrastructure Manager and the Execution

	Engine.
Method	Case study
Description	A thorough review of current monitoring and adaptation techniques reveals that most of current monitoring and adaptation techniques are very fragmented and deal with only a specific SBA layer. However, it is critical that monitoring and adaptation occur across all the SBA functional layers in order to avoid executing multiple actions. The used case study clearly emerge the need for a cross-layer approach.
Goal	The goal was to show the applicability of the proposed framework on a real world scenario.
Set-up	We used Adonis Community Edition to model our scenario of the Traffic Management Case Study.
Inputs	-
Outputs	-
Outcome	Positive
Experiences	The used case study clearly states the need for cross-layer monitoring and adaptation approaches. Our scenario can be replicated and used in other similar works.
References	Zeginis C., Konsolaki K., Kritikos K, Plexousakis D.: " *ECMAF: An Event-Based Cross-Layer Service Monitoring and Adaptation Framework*". To appear In: NFPSLAM-SOC workshop co-located with ICSOC (2011)
Glossary	Monitoring, Adaptation, Proactive Adaptation, Functional Layer of Monitoring Architecture
Keywords	Event, monitoring, adaptation, cross-layer, service, pattern, Event-Calculus

3.1.2.3 *Towards Deriving Specifications for Composite Web Services*

The approach aims to provide a thorough and efficient process of automatically deriving composite specifications based on the specifications of the participating services by attempting to deduce the minimum subset of these specifications that needs to be exposed to the service consumer. In order to do this, the composite specification is derived using structural induction, examining the composition schema using a bottom-up approach.

Validation Set-up & Result	
Name	Towards Deriving Specifications for Composite Web Services
Synopsis	The approach provides a thorough and efficient process of automatically deriving composite service specifications, which is validated by a case study.
Authors	George Baryannis, Manuel Carro, Dimitris Plexousakis
Research questions	Automatic Derivation of Composite Service Specifications
Case study	E-Government Case study: http://scube-casestudies.ws.dei.polimi.it/index.php/E-Government_Case_Study
Scenario	We use a combination of the scenarios proposed in the case study in order to represent the complete process of obtaining a government document, digitally certified or not.  In this scenario, citizens submit applications to request some government-related service, such as obtaining government-issued documents. A fee is required to

	obtain a particular document, so a mechanism that executes payment transactions is involved. Moreover, the citizen may request that the resulting document is authenticated. To that end, a digital signature certification mechanism is provided. A typical process to obtain a document is illustrated in the figure above. Users log into the system and fill in forms regarding their request as well as payment details, which are then simultaneously processed before the payment process can begin. If users demand authentication for their documents, then a certification process is executed, resulting in the delivery of a certified document to the user. Otherwise, an uncertified document is delivered.
Method	Case study
Description	Using the aforementioned case study and scenario, it becomes clear that for non-trivial composite service such as the one in the scenario, there is a need for an automated derivation process in order to obtain the composite specification that describes the complete service. The scenario was kept simple for the purposes of this work, but it could become even more complicated, in which case manually determining the preconditions and post-conditions is out of the question.
Goal	The goal was to show that the need for composite service specifications and for an automated derivation process is closely associated to the needs of real world SOA applications.
Set-up	We used the Prover9 theorem prover in the specification derivation process in order to obtain the composite specification of the process in the scenario.
Inputs	We used as input indicative service specifications for hypothetical services that implement the tasks in the scenario.
Outputs	The result is a composite specification of a service that implements the scenario of obtaining a government-issued document, with or without digital certification.
Outcome	Positive
Experiences	It became apparent, through the use of a real world case study, that composite service specifications are indispensable when it comes to correctly and completely describe the behavior of a composite service and the results of its execution.
References	G. Baryannis, M. Carro and D. Plexousakis: "*Towards Deriving Specifications for Composite Web Services*". Technical Report FORTH/ICS, Pending
Glossary	Formal Specification, Service Description, Service Composition, Service Specification, Composition Schema
Keywords	Specification, Composition

3.1.2.4 Combining SLA Prediction and Cross Layer Adaptation for Preventing SLA Violations

Validation Set-up & Result	
Name	Combining SLA Prediction and Cross Layer Adaptation for Preventing SLA Violations
Synopsis	We present an architecture as a generic framework for combining cross-layer adaptation with mechanisms for preventing SLA violation. Multiple adaptation mechanisms are available to react on adaptation needs, acting on different layers of the SBA. The final goal of the cross-layer adaptation capability is to avoid the violation of agreed Service Level (in SLAs) and thus ensure the benefits of SBAs for both customers and providers.
Authors	Eric Schmieders, Andras Micsik, Marc Oriol, Khaled Mahbub and Raman Kazhamiakin
Research questions	Supporting adaptation of service-based applications Runtime SLA Violation Prevention Cross-layer integrated and coordinated SBA adaptation mechanisms

	Means to identify adaptation strategies across layers
Case Study	N/A
Scenario	Both external and internal services are available for an SBA. Monitoring is in place for both internal and external services. The execution of service compositions is monitored, SLA violations are predicted and possible preventive solutions are calculated and the best one is enacted.
Method	Case study
Description	We present an architecture where cross-layer adaptation needs are detected, adaptation strategies and methods are automatically selected and enacted. The final goal is to avoid SLA violations.
Goal	To present the feasibility of the approach.
Set-up	Tools used: SALMon, SPADE, Jade, ESB. Theoretical validation.
Inputs	Service compositions, SLAs, service statistics
Outputs	Adaptation solutions
Outcome	positive
Experiences	N/A
References	Eric Schmieders, Andras Micsik, Marc Oriol, Khaled Mahbub and Raman Kazhamiakin. *Combining SLA Prediction and Cross Layer Adaptation for Preventing SLA Violations*. 2nd Workshop on Software Services, Timisoara, Romania, June 6-9, 2011
Glossary	Self-Adaptation, Proactive Adaptation, Service Level Agreement (SLA), SLA violation, Service-Based Application (SBA)
Keywords	Cross-layer adaptation, SLA violation prevention, cross-layer monitoring, adaptation strategies

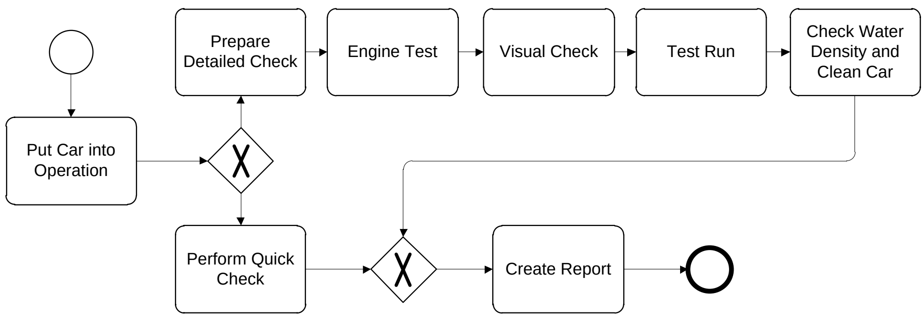
3.1.2.5 Service Network Modeling and Performance Analysis

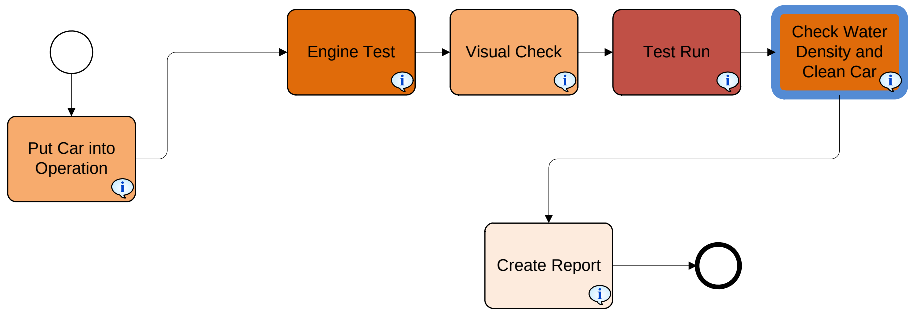
A simulation model is proposed, in order to evaluate the long-term impact of changes to resources and predict the performance of service networks.

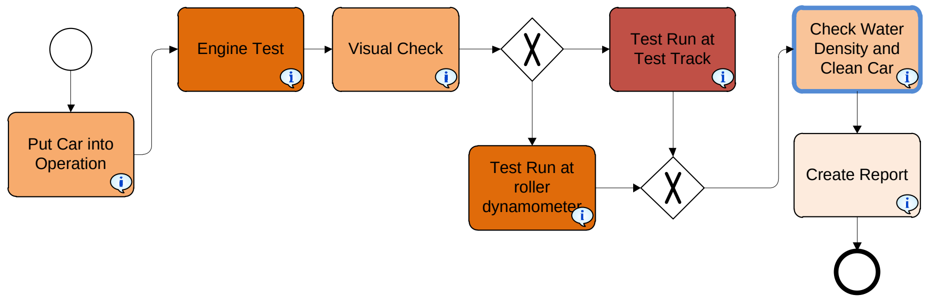
Validation Set-up & Result	
Name	Service Network Modeling and Performance Analysis
Synopsis	The approach offers a service performance analytics model based on a methodology of calculating value in service systems.
Authors	Manolis Voskakis, Christos Nikolaou, Willem-Jan van den Heuvel and Marina Bitsaki
Research questions	End-to-end processes in Service Networks
Case study	Car Repair Case Study based on the following journal publication: N.S. Caswell, C. Nikolaou, J. Sairamesh, M. Bitsaki, G.D. Koutras, and G. Iacovidis, "*Estimating value in service systems*", IBM System Journal, vol. 47, nr. 1, pp. 87-100, 2008
Scenario	OEM-franchised dealers may service and repair cars for their clients. Both activities require a car parts catalogue to ensure that repairs can be performed efficiently either in the replacement of parts or repairing after accidents. The part catalogue facilitates efficient installation, operation and lifecycle. maintenance of intricate products describing detailed part information that can be fully integrated with other service applications supporting customer support processes, human resource management, and other service provisions. The quality of the OEM parts, the OEM part catalogues, and OEM support services influences how many OEM parts will be ordered and used for a car repair and how many parts will be used from Third Party Suppliers (TPS), and how many customers will go to OEM dealers or to TPS dealers. OEM obtains parts from certified supply-chain suppliers (SCS). The technicians report the car service requirements that may

	include replacing teardowns, warranty replacements and collision repairs. On the basis of the car diagnosis, a cost estimate will be computed and communicated to the client for authorization. Once authorized the automotive technician will scrutinize failure symptoms, detect faulty parts, order parts and perform the repair. Ordering parts is a complex process that involves asking advice from expert technicians from the OEM, including acquiring information about parts under warranty, and getting approval from the dealer's part manager. The part manager then checks local inventory for the required part, and if necessary checks the stock at the OEM or supplier stocks, and eventually places an order. The part manager may either use third-party suppliers or suppliers from certified supply-chain suppliers.
Method	Case study
Description	For a car repair service network based on the above scenario, the authors offer a mechanism for calculating value that is based on a three-level hierarchical schema. In the top level, the value is generated by the combination of the values of the dealer, the suppliers, the OEM and the end customer. At the next level, the dealer's value is presented in more detail, while the third level handles the modeling of the dealer's cost. This model is analyzed via simulation of 4 variations of the car repair scenario.
Goal	The intended goal is to analyze and evaluate the proposed model for value calculation.
Set-up	Four variations of the car repair scenario are modeled and simulated: the basic one, the transformed one described above, a further scenario where dealers are replaced by new ones who offer more complementarities at the same mean repair price and a final one, that investigates Nash equilibrium strategies between OEM and the dealers.
Inputs	Rates of offerings and payment flows per month over a period of about 30 months are considered, along with a service request flow according to the Poisson distribution.
Outputs	The results show the following: (1) the transformed model offers better value for the OEM, but only after 10-12 months, (2) as the mean repair price increases, the difference between the dealers' value in the basic network and that in the transformed network is smaller, (3) a change in the network that improves its performance, such as new dealers who offer more complementarities, may affect positively some participants and negatively others and (4) when the dealer increases its profit rate, OEM's optimal choice is to decrease its optimal profit rate. Conversely, if OEM increases its profit rates, the dealer optimally decreases its profit rate.
Outcome	Positive
Experiences	
References	Manolis Voskakis, Christos Nikolaou, Willem-Jan van den Heuvel and Marina Bitsaki, "*Service Network Modeling and Performance Analysis*", The Sixth International Conference on Internet and Web Applications and Services ICIW 2011 March 20-25, 2011 - St. Maarten, The Netherlands Antilles M.Bitsaki, C. Nikolaou, M.Voskakis, Willem-Jan van den Heuvel, K. Tsikrikas: *Performance Analysis and Strategic Interactions in Service Networks*. Submitted in Journal On Advances in Networks and Services (Accepted and will be published on February).
Glossary	Agile Service Network, Performance
Keywords	Service networks, value optimization, performance analysis

3.1.2.6 Architecture and Methodology for Managing the KEI of Business Processes

Validation Set-up & Result	
Name	Architecture and Methodology for Managing the KEI of Business Processes
Synopsis	The approach provides an architecture and methodology for realizing a sustainable business process reengineering. The approach is evaluated by a case study.
Authors	Alexander Nowak, Frank Leymann, David Schumm, Branimir Wetzstein
Research questions	Managing the Key Ecological Indicators of Business Processes
Case study	Case study from a car manufacturer.
Scenario	We elaborate a car finishing process of a car manufacturer. This process is performed every time a new car has been assembled and leaves the assembly line. In the first step of the finishing process the car is put into operation making sure all systems are working. Then, in the regular case a quick check based on a predefined checklist is performed. In some cases a detailed check is performed. This part of the process first includes the transportation of the car to the test center and the preparation of the test procedure. The test procedure then starts with an engine test which is followed by a detailed visual check of interior and exterior. After the test run on a test track in the next process step, the water density is checked and the car gets cleaned and prepared for delivery or refinishing, respectively. Finally, in both cases a detailed report of the test results is created and sent to the operations management. Performing the finishing process either with a quick or a detailed test run results in different cost, quality, and duration characteristics of the complete process depending on the specific weights of those dimensions, e.g. the percentage of detailed tests that can be managed. 
Method	Case Study
Description	Based on our initial process model we applied the proposed methodology in order to identify potential adaptation strategies. In a first step we have performed different steps: (1) we have omitted all activities that cannot be changed at all. (2) we augmented the business process activities with their corresponding KEI. The result is shown in the following figure:

<table>
<tr><td></td><td>

In a next step we have created a process variant that reduces the total environmental impact of this business process:


Within the new case, the costs will slightly increase due to the higher costs of a test run at the roller dynamometer test bench. On the other side, we will save a significant amount of time for not transporting the car to the test track and the more efficient test run. What is important now is that we can also achieve savings at the water consumption and CO2 emission. Based on the weights strategically set for those four dimensions we can try to determine the percentage of tests which should go to the new roller dynamometer test activity and configure the branching activity accordingly.</td></tr>
<tr><td>Goal</td><td>The Goal was to show how Key Ecological Indicators can be used by organizations in order to optimize the environmental impact of their business processes.</td></tr>
<tr><td>Set-up</td><td>For evaluating the architecture and methodology of the approach we set-up a case study that represent a business process from a car manufacturer. The proposed methodology was applied to this business process in order to identify the environmental impact, visualize this impact, and propose a process variant with less environmental impact in terms of CO2 emission and water consumption.</td></tr>
<tr><td>Inputs</td><td>A BPMN Process.</td></tr>
<tr><td>Outputs</td><td>The Output was twofold: (1) A BPMN process view that highlights the parts of the process relevant to the environmental impact of the organization. (2) A process variant that provides a possible process adaptation strategy.</td></tr>
<tr><td>Outcome</td><td>Positive.</td></tr>
<tr><td>Experiences</td><td>We are confident that the proposed approach provides a good starting point for organizations to analyze their business processes in terms of their environmental impact.</td></tr>
<tr><td>References</td><td>Nowak, Alexander; Leymann, Frank; Schumm, David; Wetzstein, Branimir: An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering. In: Proceedings of the 1st International Conference on ICT as Key Technology for the Fight against Global Warming - ICT-GLOW 2011 (2011).</td></tr>
</table>

Glossary	Key Performance Indicator, Monitoring, Adaptation, Analysis, Service Composition, Business Process
Keywords	Business Processes, Process Views, Process Monitoring, Adaptation, Environmental Impact, Green Business Process Reengineering

3.1.2.7 Design for Self-adaptation in Service-oriented Systems in the Cloud

In this work we go through the S-Cube lifecycle highlighting activities that are needed to support adaptation. Adaptation can be performed at various layers of the service-oriented system. In particular, it can concern the layer where services are composed together or the layer of the executing infrastructure, typically, a cloud system. To exemplify the various steps and activities of the lifecycle we focus on the fact that the Infrastructure-as-a-Service approach enables the development of Real-Time Online Interactive Applications (ROIA), which include multi-player online computer games, interactive e-learning and training applications and high- performance simulations in virtual environments. We illustrate how the Lifecycle Model expresses the major design and execution aspects of ROIA on Clouds.

Validation Set-up & Result	
Name	Design for Self-adaptation in Service-oriented Systems in the Cloud
Synopsis	We describe how the generic Lifecycle Model developed in the S- Cube project for the design and management of Service Based Applications (SBA) can be utilized in the context of Cloud Computing.
Authors	Antonio Bucchiarone, Cinzia Cappiello, Elisabetta Di Nitto, Sergei Gorlatch, Dominique Meiländer, Andreas Metzger
Research questions	Design of adaptive Service Based Applications (SBA) in Cloud Computing.
Case study	Multi-player action game called RTFDemo (www.real-time-framework.com)
Scenario	In ROIA, there are typically multiple users who access a common application state and interact with each other concurrently within one virtual environment. The users access the application from different client machines and control their avatars that interact with other users' avatars or computer-controlled characters. Since ROIAs have very high performance requirements, the application state processing is performed on multiple servers; the virtual environment is divided into disjoint areas (zones) and each server is processing clients inside a particular zone. Hence, ROIA are highly distributed applications with challenging QoS demands, such as: short response times to user actions (about 0.1-1.5 s), high update rate of the application (up to 50 Hz), large and frequently changing number of users in a single application instance (up to 104 simultaneously). For ROIA, a particular challenge is that the number of users participating in a ROIA session and thus, the workload, is often subject to daytime-dependent changes. A negative consequence of this are expensive up-front investments to build a suitable server pool which is able to handle peak user numbers but will be underutilized most of the time when the load is below the peak. Hence, dynamic adaptation of application sessions during runtime is crucial for ROIA. We address this adaptation challenge using Cloud Computing to provide additional resources. Cloud Computing allows to add/remove resources on demand and promises a potentially un- limited scalability by distributing frequent state computations on an arbitrary number of resources given suitable adaptation mechanisms. Despite a variable number of users, Cloud resources can be used efficiently if the application provides suitable adaptation mechanisms. Hence, using Cloud Computing for resource provision and the Lifecycle Model for implementing adaptable ROIA complement each other.

Method	Case study
Description	The S-Cube lifecycle for the design of adaptive Service-based application is analyzed in order to highlight those activities that are needed to support adaptation. The adaptation activities can be performed at various layers of the service-oriented system. In particular, they can concern the layer where services are composed together or the layer of the executing infrastructure, typically, a cloud system. To exemplify the various steps and activities we focus on the fact that the Infrastructure-as-a-Service approach enables the development of Real-Time Online Interactive Applications (ROIA), which include multi-player online computer games, interactive e-learning and training applications and high-performance simulations in virtual environments. The work shows how the Lifecycle Model expresses the major design and execution aspects of ROIA on Clouds by addressing the specific characteristics of ROIA: a large number of concur- rent users connected to a single application instance, enforcement of Quality of Service (QoS) parameters, adaptivity to changing loads, and frequent real-time interactions between users and services. Note that in order to support ROIA development and adaptation on Clouds, a RTF-RMS resource management system on top of the Real-Time Framework (RTF) has been developed. RTF-RMS implements the following mechanisms for ROIA development on Clouds: **1.** *Monitoring* of application-specific data, e.g., update rate, number of entities, etc. **2.** *Distribution handling* for the dynamic adaptation of application sessions by adding/re-moving Cloud resources using particular adaptation strategies. **3.** *Application profiles* for implementation of application-specific adaptation triggers. 4. High-level *development support* for communication handling and application state distribution.
Goal	The goal was to show how the Lifecycle Model expresses the major design and execution aspects of Service-Based Applications in a cloud system.
Set-up	Cloud environment with the Eucalyptus framework (version 2.0.2); servers are Intel Core Duo PCs with 2.66 GHz and 2 GB of RAM running CentOS 5.3.
Inputs	Service-based application analysis: definition of functional, non-functional and adaptation requirements.
Outputs	Identification of self-adaptation triggers and strategies that are relevant for the specific system under development and support in the design of the proper runtime monitoring and reasoning mechanisms needed to execute these adaptation strategies.
Outcome	Positive
Experiences	The life cycle offered by the S-Cube project is a helpful tool to support SBAs designers.
References	D. Meiländer, A. Bucchiarone, C. Cappiello, E. Di Nitto, S. Gorlatch, *Using a Lifecycle Model for Developing and Executing Real-Time Online Applications on Clouds,* In proc. of WESOA 2011. A. Bucchiarone, C. Cappiello, E. Di Nitto, S. Gorlatch, D. Meiländer, A. Metzger, *Design for Self-adaptation in Service-oriented Systems in the Cloud,* In: European Research Activities in Cloud Computing, 2012.
Glossary	Adaptable Service-based Application, Adaptation requirements and objective, Adaptation strategy
Keywords	Adaptation, Service-based applications, Cloud computing

3.1.3 Validation Method: Field Study

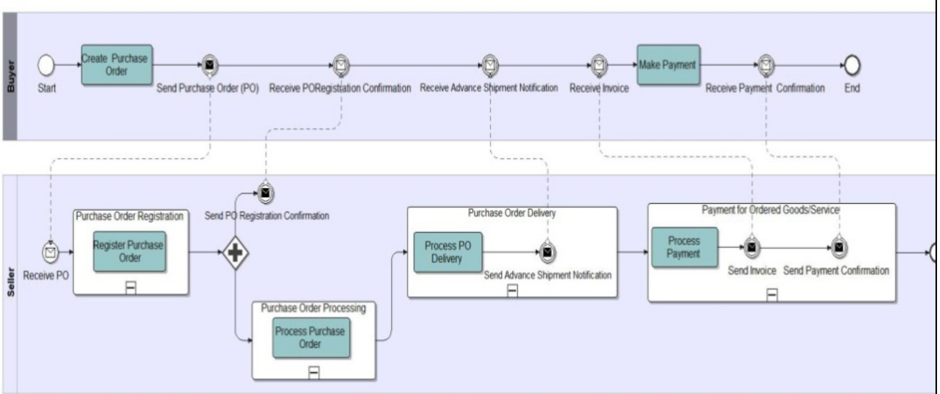
No experiment has been reported under this category.

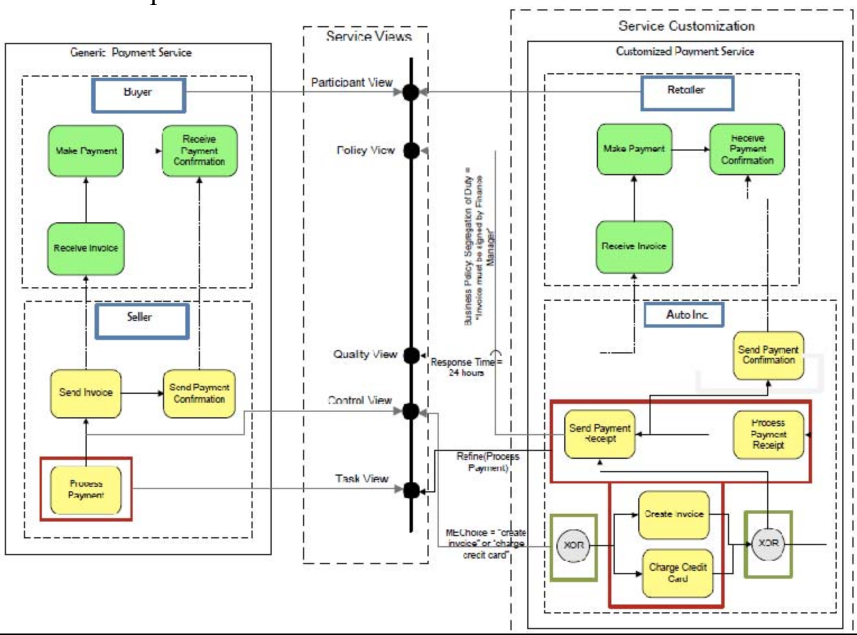
3.1.4 Validation Method: Action Research

3.1.4.1 Customization of business services at design-time to build context specific service based applications(SBAs)

This method of service customization has been emerging steadily as a concept as the popularity of service oriented computing technologies increases (e.g., WSDL, BPEL, and so on). Earlier research into service customization focused on configurable EPC approach that is limited to service functions, which is obviously important but not adequate for the modern service-driven business environment. The service-driven business environment of today also involves diverse non-functional requirements including quality-of-service, business policy, and security aspect. Customizing services covering such a wide variety of requirements from disparate domains is a non-trivial task and organizations hire experts to perform these tasks, which increases development costs. This implies that a suitable approach is paramount to enable users without having expertise on Information Technology (IT) to customize both functional and non-functional aspects of services.

Validation Set-up & Result	
Name	Customization of business services at design-time to build context specific service based applications(SBAs)
Synopsis	The approach eases the complexities of service customization and allow non-IT experts (e.g. business analysts) without a background in service related technologies (e.g. BPEL) to customize services to building context specific SBAs
Authors	Yehia Taher, Rafiqul Haque, Michael Parkin, Willem-Jan van den Heuvel, Ita Richardson, Eoin Whelan
Research questions	• Customization of quality view of services using parameters specific to context • Customization of policies of services specific to context • Customization of tasks that carry out service operations • Customization of control flows
Case study	Automotive case study from here: http://scube-casestudies.ws.dei.polimi.it/index.php/Main_Page
Scenario	We use a small portion of "purchase order management"- end-to-end business process that is based on the electronic commerce domain. This portion we call purchase order process. We then generalize this process through *abstracting* the services (activities) contained in the process to make it a *reusable* process. The generalized process of purchase order is depicted below.



<table>
<tr><td></td><td>The blue rectangular boxes denote the task or operations (aka atomic activity) that can be composed to a service (aka sub-process, activity, business function, and sometime process) shown using white rectangle box. In the example, reusable services are generic contained in service repository provided by third party service providers. In process diagram, the two pools represent the partners (Buyer and Seller) involved in the purchase order process. No specific partner name is given as the process is a generic. The process contains generic activities including register purchase order, process purchase order, process purchase order delivery and process payment. These activities are generic because they are captured from global point of view, i.e., these are activities commonly used by selling organizations. From the perspective of buying organizations, the two generic activities involved in order management are purchase order creation and make payment. The commonality has also shown in flow of messages between buyer and seller. Besides, the business logic that describes the order of activities has been abstracted (generalized) as well. These generic features of this process facilitate the business organization to reuse it.</td></tr>
<tr><td>Method</td><td>Action Research</td></tr>
<tr><td>Description</td><td>Services designed from global or common perspective cannot be reused directly in a specific context. However, what exactly are the factors that preclude the direct reusability of generic services? The simple answer is contextual requirements that vary as the circumstances the service is used in change. For example, the delivery service (shown in process diagram) will vary among business types, organizations and the locations it is used in. Another example is the payment service which relies on business-specific policies. For example, in business-to-business (B2B) scenarios, buying companies in Europe must issue a letter of credit to sellers in South-east Asia before the order is processed, with the letter of credit a legal confirmation from the buyer to credit a certain amount from his account to the seller’s account. However, this may not be required for buyers from the United States. The requirements can be more diverse in case of business-to-consumer (B2C) and are enormously important because they are the driving factors that ensure customer satisfaction and provide a competitive advantage for businesses. As discussed above, context-independent reusable services mostly do not adopt these requirements because they are specific to certain context. Apart from the non-functional perspective, the functional requirements are also important and vary from context to context. Thus, the functional features of a generic service also may not satisfy the requirement of specific context

</td></tr>
</table>

	These considerations give the rise to the need of *customization* for fine-tuning generic services at design-time so that they may be reused to building SBAs specific to a context.To cater support for customization of services, a suitable approach that eases customization is strongly required. The customization approach must simplify customization complexities to enable a non-IT expert to customize services. The figure shows how a generic process (in the leftmost block) is customized to context specific process (the rightmost block). The block with dotted line (in the middle) is the service view segmentation layer of proposed solution. A generic process is fed into this layer as input, and then the service view mapper maps the process to five different views entail participant view, policy view, quality view, control view, task view(see above figure). The subsequent service customization layer provides the customization parameters (or attributes). In above figure for instance, response time is a parameter of quality view whose value is 24 hours assigned by the user.
Goal	The goal was to show the ease of use of the proposed multilayered customization approach. The validation should support the claim that the proposed approach simplifies the customization of services by providing step by step guidelines to customize services to build context specific service-based applications.
Set-up	For evaluating the proposed customization approach, we take a small portion (process) from end-to-end purchase order management process of automotive case study. We then redesign that process to turn it into a generic process. We invite a business personnel having basic knowledge on information technology. We provide him the generic process, a documentation of specific requirements for a business process, a sketch-board and the technical specification of our customization approach.
Inputs	We use a generic process, which we designed in Business Process Modeling Notation (BPMN). We use a documentation of Quality of Service requirements for business process of a specific organization.
Outputs	First, using the proposed customization approach, at service-view segmentation layer, the user successfully maps the process into different views. Second, the user selects parameters provided in the specification and assigns the value for each parameter, which successfully leads to building a business process for a specific organization.
Outcome	Positive
Experiences	The user was little uncomfortable in the beginning since the solution was provided on paper. According to him, it would have been easier for him to carry out actions if the solution would be prototype based. After finishing all tasks, the user was very much confident that the solution was very much supportive to him guiding the customization of process although he had not much idea on technical aspects. He added that the multi-layer approach gave him the 3600 degree view of service customization and the customization parameters provided by the approach do most of his works but only assigning required values of those parameters – a task which is specific to a context - can be done easily by anyone who even do not have much knowledge on business process. We strongly believe that the solution will be very useful within the domain of "Internet of Things". It will facilitate integrating the on-demand services just in time within running application. We also believe that, the positive externality of this approach would be reduction of service customization cost as well as development cost since there will not be any expert necessary to carry out tasks to customize service to building SBAs.
References	Yehia Taher, Rafiqul Haque, Michael Parkin, Willem-Jan van den Heuvel, Ita

	Richardson, Eoin Whelan: *Towards proactive adaptation: A journey along the s-cube service lifecycle.* In: Maintenance and Evolution of Service-Oriented Systems (2010)
Glossary	Service, Reusability, Customization, Just in time integration, Service Oriented Computing (SOC), Service Based Application (SBA), design-time.
Keywords	Service, Reusability, Customization, Service Oriented Computing (SOC), Service Based Application (SBA)

3.1.5 Validation Method: Survey

3.1.5.1 A Survey of SOA Migration in Industry

Validation Set-up & Result	
Name	A Survey of SOA Migration in Industry
Synopsis	This research provides deeper understanding of how migration is performed in industrial practice. In essence, the research contrasts industrial migration approaches and academic ones. By discussing differences, we obtained promising directions for industry-relevant research.
Authors	Maryam Razavian, Patricia Lago
Research questions	What methods/approaches regarding legacy to SOA migration are used in practice?
Case Study	In this research we conducted an industrial interview survey in seven leading SOA solution provider companies With the objective of understanding the industrial migration approaches, we designed and executed the interviews. Each interview was analyzed considering the constituent conceptual elements of a migration process as proposed in [1], including the activities carried out, the available knowledge assets, and the overall organization of migration process. Furthermore, we looked for the best practices that companies have developed out of experience for successful legacy migration.
Scenario	N/A
Method	We chose survey as our research method for two main reasons. First "what questions" are likely to favor surveys. Second, due to explorative nature of this research we looked for multiple data points. Surveys again are a suitable method for this purpose as they address larger target population compared to other research methods such as case studies.
Description	As a result we found that, in fact, all companies converge to the same, one, common SOA migration approach. This suggests that industrial migration approaches converge to a similar set of activities, process organization, and best practices, in other words, with experience enterprises mature toward a similar approach to SOA migration. In addition, we contrasted the industrial approaches with academic ones, which we identified from a previous Systematic Literature Review (SLR) on SOA migration [2].
Goal	a. To gain a deeper understanding of the types of migration approaches in industrial practice. b. To identify the gaps between industrial migration approaches and academic ones
Set-up	To gather information about industrial SOA migration approaches, we conducted a series of interviews. Interviews are an appropriate strategy when the goal is to identify the experience of the individual and/or organizations in carrying out a task. The interviews conducted were semi-structured interviews. The open-ended questions of this type of interviews allow interviewers to ask follow-on questions when necessary. The first version of the interview guide was piloted with one researcher and one practitioner with experience in SOA migration 1. The theoretical population target of this study included architects (both technical and

	enterprise architects) with considerable experience in carrying out SOA migration projects. To recruit suitable architects we created a leaflet illustrating the goals of this study and distributed it in the Dutch architecture conference 2009, a well-known practitioner conference with over 500 participants each year. Finally nine architects, affiliated with the international companies residing in the Netherlands and Belgium committed to our study. To enhance the reliability of the result we carefully chose the companies that are leading SOA solution providers. To ensure that the industrial survey is based on real-world experience, instead of participant's opinion on how migration should be carried out, we asked them to select one recent project and answer the questions considering that specific project. Before carrying out the interviews, the interviewees were sent a copy of the interview guide as well as some background information on the study. This allowed us to better synchronize terminology. The interviews were conducted on site and were all video-recorded. The initial findings along with any remaining unanswered questions were iterated with the interviewees to ensure that we understood correctly, and that nothing was missing.
Inputs	1. The SOA Migration Framework presented in [1] 2. The results of systematic review presented in [2]
Outputs	Results show that by supporting similar set of activities, process organization, and best practices, industrial migration approaches do converge to one, common, type of migration. As such, this paper suggests that the industrial approaches mature towards a similar approach to SOA migration. This approach is driven by common types of knowledge elements and core activities, fundamentally driven by To-Be state, and with little to no attention to reverse engineering. Further findings show that industrial approaches, strictly follow incremental migration.
Outcome	positive (the research result fulfills its goal)
Experiences	-
References	[1] M. Razavian and P. Lago , "Towards a Conceptual Framework for Legacy to SOA Migration". In: Fifth International Workshop on Engineering Service-Oriented Ap- plications (WESOA'09). (2010) 445-455 [2] M. Razavian and P. Lago, "A Frame of Reference for SOA Migration," in Towards a Service-Based Internet, ser. Lecture Notes in Computer Science, vol. 6481, 2010, pp. 150–162. [3] M. Razavian and P. Lago, "A Survey of SOA Migration in Industry," in International Conference on Service Oriented Computing, ICSOC, 2011.
Glossary	References to relevant terms in the knowledge model.
Keywords	SOA Migration; Migration Strategies

3.1.6 Validation Method: Literature review

3.1.6.1 Towards Classification Criteria for Process Fragmentation Techniques

Validation Set-up & Result	
Name	Towards Classification Criteria for Process Fragmentation Techniques
Synopsis	The approach provides an architecture and methodology for realizing a sustainable business process reengineering. The approach is evaluated by a case study.
Authors	Michele Mancioppi, Olha Danylevych, Dimka Karastoyanova, Frank Leymann
Research	How to classify fragmentation techniques for business processes

questions	
Case study	N/A
Scenario	N/A
Method	Literature review
Description	We investigated classification criteria for process fragmentation techniques based on the "seven Ws", namely Why, What, When,Where, Who, Which, and hoW. The presented classification criteria are applied to some of the process fragmentation approaches available in the literature.
Goal	Elicit criteria for classifying process fragmentation techniques in order to simplify comparison and selection of process fragmentation techniques, and highlight gaps in the state of the art
Set-up	N/A
Inputs	A selection of works proposing process fragmentation techniques
Outputs	A set of classification criteria for process fragmentation technique
Outcome	Positive
Experiences	The elicited criteria seem to provide a solid foundation for classifying process fragmentation techniques. They may be further refined for classes of process fragmentation techniques such as those targeted at enabling distributed execution of processes.
References	Nowak, Alexander; Leymann, Frank; Schumm, David; Wetzstein, Branimir*: An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering.* In: Proceedings of the 1st International Conference on ICT as Key Technology for the Fight against Global Warming - ICT-GLOW 2011 (2011).
Glossary	Business Process, Business Process Fragmentation
Keywords	Business Processes, Process Fragmentation

3.1.7 Validation Method: Prototyping

3.1.7.1 Evaluation of user task models in service discovery

JRA1.1 sought to introduce human-computer interaction knowledge into the design and use of service-based applications. Earlier research had identified user task knowledge as one form of human-computer interaction knowledge that can inform the design of service-based applications more effectively. This result reports the result of a large experiment to investigate the effect of reusing user task models during a service discovery activity. Results revealed that introducing user task models increased the effectiveness of service discovery over basic discovery engines, but not over other sophisticated engines.

Validation Set-up & Result	
Name	Evaluation of user task models in service discovery
Synopsis	Matched user task models expressed using CTTs were made available to reformulate service queries then fired at a large service registry. Retrieval results were used to explore the relative effectiveness of service query reformulation with knowledge from user task models.
Authors	Konstantinos Zachos, Angela Kounkou and Neil Maiden
Research questions	The evaluation research investigated the following four hypotheses: **H1**: Extending rather than replacing service queries with additional knowledge about user tasks will *improve the overall effectiveness of service discovery;* **H2**: The reformulation of service queries with knowledge about user tasks will *decrease the number of irrelevant services* retrieved by a service discovery engine; **H3**: The reformulation of service queries with knowledge about user tasks will

	increase the number of relevant services retrieved by a service discovery engine; **H4**: The reformulation of service queries with knowledge about user tasks will *improve the overall correctness of services* retrieved by a service discovery engine.
Case study	The e-government domain.
Scenario	Service queries used to evaluate the approach were drawn from the e-government domain. The target service registry for both evaluations was a version of a SeCSE service registry containing descriptions of 215 existing web services in domains ranging from flight booking and weather reporting to route planning. We populated the user task model catalogue with codified user task models relevant to the evaluation. Seven were codified for tasks that most citizens would be expected to undertake from time to time: • *Obtain health advice* • *Access a patient record* • *Request a route* • *Calculate a distance* • *Park a car* • *Zoom into a map* • *Obtaining a weather forecast*
Method	Prototyping
Description	The reported results were used to accept or reject each of the four hypotheses under investigation: **H1** That extending rather than replacing service queries with additional knowledge about user tasks will *improve the overall effectiveness of service discovery* was **accepted** because the balanced F-score measures for *extend* strategies were significantly higher than for *replace* strategies; **H2** That the reformulation of service queries with knowledge about user tasks will *decrease the number of irrelevant services* retrieved by a service discovery engine was **rejected** because the mean average precision measures generated with TEDDiE were lower than the equivalent measures with EDDiE; **H3** That the reformulation of service queries with knowledge about user tasks will *increase the number of relevant services* retrieved by a service discovery engine was **partially accepted** because the recall measures generated generated with TEDDiE-Lite were significantly higher than the equivalent measures with EDDiE-Lite. However this was not the case between Full-TEDDiE and Full-EDDiE; **H4** That the reformulation of service queries with knowledge about user tasks will *improve the overall correctness of services* retrieved by a service discovery engine TEDDiE will improve the overall correctness of services retrieved was **partially accepted** because the balanced F-score measures for TEDDiE-Lite were significantly higher than for EDDiE-Lite. Again however this was not the case between Full-TEDDiE and Full-EDDiE.
Goal	As stated above, the purpose of the evaluation was to investigate the effective of reusing user task knowledge on service discovery
Set-up	The set-up was a controlled study to evaluate the effect of different treatments on service discovery. We fired a total of 92 service queries at the target service registry. The number of generated EDDiE service queries was lower than for TEDDiE because of the presence of multiple sub-tasks in user task models used to generate multiple service queries. Each *discovered service* was compared manually against the list of *classified relevant services* for the use case specification from which each service query was generated. If the service was on the list, then the service was categorized as a *retrieved relevant service*. The totals of *classified relevant services*, *discovered services* and *retrieved relevant services* where then analyzed using established measures of precision and recall from information retrieval research. Using these measures the precision for each service query was defined as: *(Total retrieved relevant services / Total discovered services)*100*

	The recall for each service query was defined as: *(Total retrieved relevant services / Total classified relevant services)*100* Finally the balanced F-score for each service query was defined as: *(2*Precision*Recall / Precision + Recall)*100* One precision measure, one recall measure and one balanced F-score measure was generated for each of the 92 service queries fired at the service registry, providing a total of 276 precision, recall and balanced F-score measures. Of the 92 service queries 88 were generated by TEDDiE and 4 by EDDiE, and 46 were generated by Full-EDDiE and Full-EDDiE and 46 TEDDiE and TEDDiE-Lite. Results are reported in the next section.
Inputs	A total of 92 service queries
Outputs	The outputs were the totals of retrieved *relevant* and *irrelevant* services for each of the 92 service queries and 276 computed precision, recall and balanced F-score measures to investigate each hypothesis in turn.
Outcome	Evidence of positive effect of reuse of user task knowledge on service discovery over basic service discovery algorithms, but not over more sophisticated service discovery algorithms.
Experiences	Given that generating and codifying user task knowledge in models has a cost associated with it not incurred with available on-line thesauri, and that service queries are not restricted to tasks that instantiate the class-level user task models in a catalogue, the reuse of user task models needs to offer more capabilities than implemented in TEDDiE.
References	K. Zachos, A.Kounkou, and N.A.M. Maiden, *Exploiting Codified User Task Knowledge to Discover Services,* submitted to IEEE Transactions on Service Computing.
Glossary	Service discovery, user task knowledge, user task models, CTT, service retrieval measure, service recall measure
Keywords	Service discovery, user task knowledge, user task models, CTT, service retrieval measure, service recall measure

3.1.7.2 Chemical Based Middleware for Workflow Instantiation and Execution

The approach aims at developing a chemical-based middleware infrastructure for dynamic workflow instantiation and execution providing a QoS-aware selection and binding mechanism at two different levels: the service composition and service enactment level.

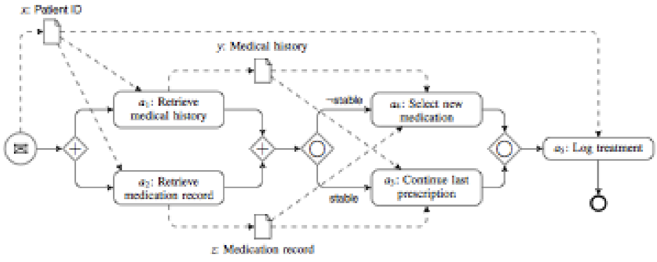
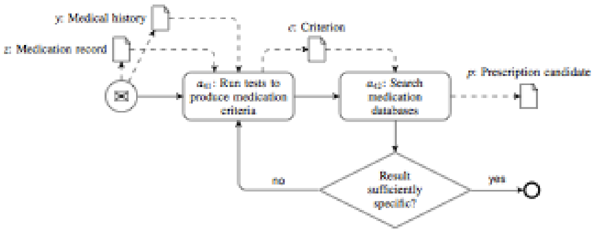
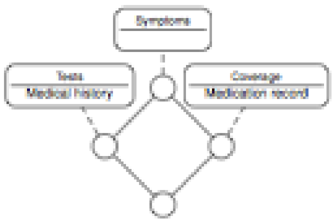
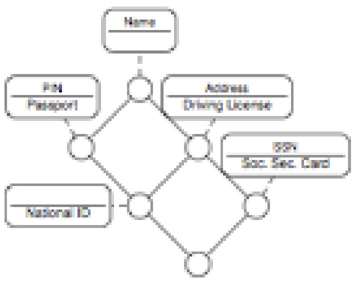
Validation Set-up & Result	
Name	Chemical Based Middleware for Workflow Instantiation and Execution.
Synopsis	A chemical-based middleware infrastructure for dynamic workflow instantiation and execution providing QoS-aware selection and binding mechanism at two different levels: the service composition and service enactment level.
Authors	Di Napoli, C., Giordano, M., Pazat, J-L., Wang, C.
Research questions	Supporting adaptation of service-based applications
Case study	Not Present.
Scenario	The scenario is a market-based service provisioning where different providers submit *service offers* consisting of actual service implementation references with quality level specification, and users submit SBA requests with a quality level requirement for the entire application.
Method	Prototyping
Description	The run-time service selecting and binding mechanisms are implemented

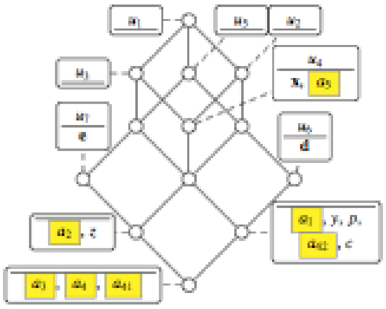
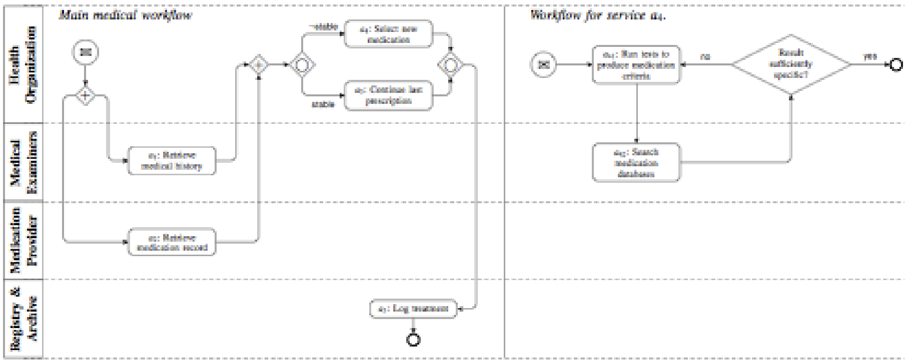
	according to the chemical computing model so that service instantiation and execution can be modeled as an evolving and autonomic system. The mechanisms occur at two levels: the first at the composition level and it allows to build a fully instantiated workflow taking into account global QoS requirement. The second occurs at the service binding level and it allows to select a qualified concrete service for each execution activity to guarantee the global QoS compliance.
Goal	The middleware aims at providing the following desirable properties: 1) all partner services are selected and bound dynamically for each execution; 2) client's QoS requirements are taken into account in the selection of services, and 3) the execution of a workflow is decentralized.
Set-up	Implementation of the middleware in the HOCL chemical programming environment developed by INRIA.
Inputs	A synthetic test-bed consisting of an SBA request in terms of a chemical abstract workflow with QoS requirements and a set of service offers.
Outputs	Verification of the middleware expected behaviour on the synthetic test-bed.
Outcome	Positive.
Experiences	We experienced that the chemical programming model can be successfully used to design and implement autonomic characteristics necessary for service selection and binding mechanisms when dynamicity in QoS provision has to be taken into account in a market-based service provisioning reference scenario.
References	C. Di Napoli, M. Giordano, J-L Pazat, C. Wang: *A Chemical Based Middleware for Workflow Instantiation and Execution*. ServiceWave 2010, pp. 100-111.
Glossary	Service binding, quality attributes, adaptability, workflow execution.
Keywords	QoS-aware service selection, chemical programming, workflow execution.

3.1.7.3 *Automated attribute inference in complex service orchestrations based on sharing analysis*

The user specifies attributes for the input data as a Boolean matrix, and the sharing-based analyzer returns the augmented matrix that includes other data items and activities.

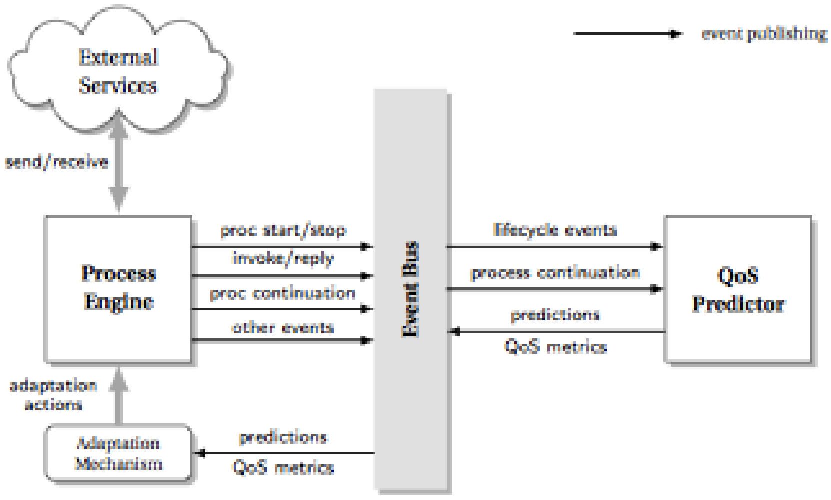
Validation Set-up & Result	
Name	Automated attribute inference in complex service orchestrations based on sharing analysis.
Synopsis	The approach above was validated using a prototype analyzer that operates on a service orchestration translated to the form of a Horn-Clause (i.e., logic) program. The inputs and outputs were visualized using lattice visualization tools.
Authors	Dragan Ivanovic (UPM), Manuel Carro (UPM, IMDEA Software), Manuel Hermenegildo (UPM, IMDEA Software)
Research questions	– Service fragmentation and merging – Analyzing QoS of service compositions
Case Study	E-Health Case Study: http://scube-casestudies.ws.dei.polimi.it/index.php/E-Health_Case_Study
Scenario	A simplified process for medicine prescription to patients of a health organization.

	 Fig. 1. An example drug prescription workflow.  Fig. 2. Selection of new medication.
Method	Prototyping
Description	A simplified workflow for medication prescription was analyzed with respect to user-defined attributes that describe information content of input data, to deduce attributes shared between the these inputs and outputs / intermediate data items and activities in the workflow. The result was used to split the process into fragments that can be executed in different organizational settings without leaking information.
Goal	The goal was to demonstrate applicability of the sharing-based attribute inference for splitting a centralized workflow into process fragments that can be executed in a decentralized manner, based on user-defined data attributes that, in this case, model data privacy / confidentiality in an information flow.
Set-up	The set-up included preparation of the workflow in the form of Horn-Clause program that is amenable to sharing analysis.
Inputs	The input was the concept lattice describing attributes of input data:  (a) Concept lattice for medical databases.  (b) Concept lattice for identity documents.
Outputs	The outcome consisted of: (1) The resulting concept lattice where input data attributes are related to the intermediate data items and activities in the workflow:

	 Fig. 9. The resulting concept lattice. and (2) fragmentation of the process based on the attributes:  Fig. 11. An example fragmentation for the drug prescription workflow.
Outcome	Positive.
Experiences	Development of tools for automated translation of workflow structure into Horn-Clause form can greatly assist the applicability of this approach.
References	Ivanović, D., Carro, M., & Hermenegildo, M. (2011). *Automated Attribute Inference in Complex Service Workflows Based on Sharing Analysis*. In Proceedings of the 8th IEEE Conference on Services Computing SCC 2011. IEEE Press.
Glossary	Service fragmentation Data-Aware Service Composition
Keywords	workflow; business process; service composition; horn clause; static analysis

3.1.7.4 *Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations*

To predict SLA violation or compliance for a running instance of a service orchestration (e.g., for running time), we use a predictor that accepts the orchestration continuation (the description of what remains to be done in the orchestration from the given point in execution until the end) and assumptions on QoS metrics of component services, given as ranges of values. From these, we automatically construct a constraint model and solve it against the given SLA objective. The solutions tell if both or just of the two cases are possible, and under which structural parameters, such as paths taken in branching or the number of loop iterations.

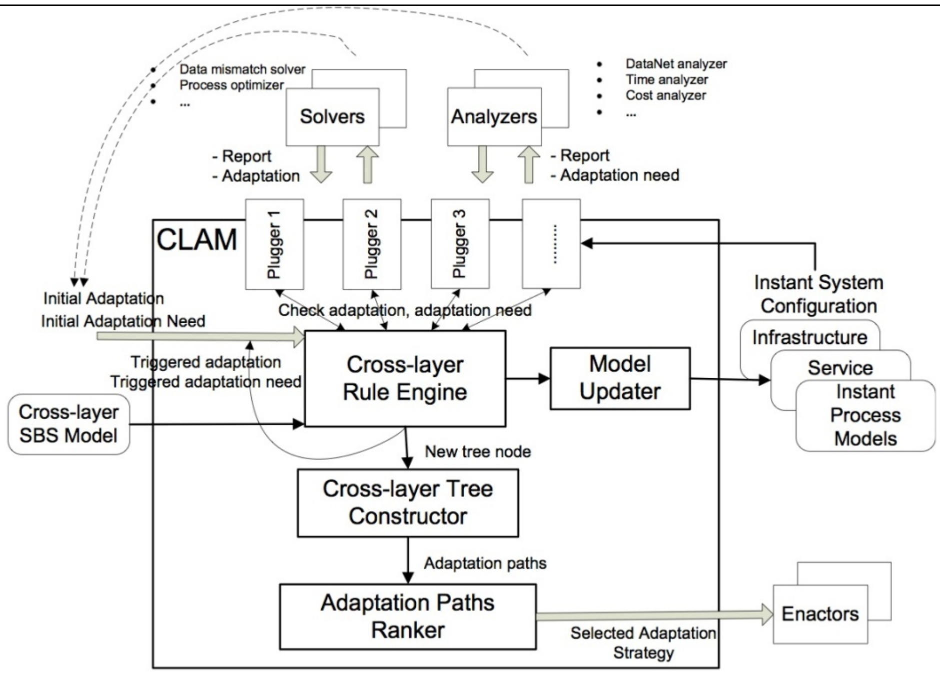
Validation Set-up & Result	
Name	Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations
Synopsis	The constraint-based approach for runtime prediction of SLA violations was validated by analyzing a sample workflow for reconfiguration of audio/video streaming content.
Authors	Dragan Ivanovic (UPM), Manuel Carro (UPM, IMDEA Software), Manuel Hermenegildo (UPM, IMDEA Software)
Research questions	– Analyzing QoS of service compositions – Predicting QoS for service compositions
Case study / Scenario	Custom. We assume a simple setting where a provider of streaming audio/video content periodically runs reconfiguration to adjust the offered content with the user's usage pattern.
Method	Prototyping. We have developed the analyzer that accepts component service metric range descriptions and the process continuations and produces the prediction of SLA violation/compliance.
Description	The validation demonstrated that the constraint-based QoS prediction approach can provide useful information ahead of failure that not only indicates the possible outcome (SLA violation and compliance), but also relate them to circumstances under which these occur (taking different paths during execution, number of iterations, etc.).
Goal	To illustrate applicability of the approach.
Set-up	We used a test architecture for constraint-based QoS prediction: 
Inputs	(1) The orchestration constraint continuation for deriving the constraint model. (2) Assumptions on running time of the component services.
Outputs	Predictions and their timing for different running time SLA limits.
Outcome	Positive.
Experiences	The initial predictions (at the start of the orchestration) are informative, and successive predictions taken as the execution unwinds tend to become more accurate.
References	Ivanović, D., Carro, M., & Hermenegildo, M. (2011). *Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations*. In G. Kappel, H. Motahari, & Z. Maamar (Eds.), Service-Oriented Computing – ICSOC 2011. Springer Verlag.
Glossary	Service Orchestrations, Quality of Service, Service Level Agreements, Monitoring, Prediction, Constraints.

Keywords	Service Orchestrations, Quality of Service, Service Level Agreements, Monitoring, Prediction, Constraints.

3.1.7.5 Cross-Layer Adaptation of SBAs

Adaptation is a cross- and multi-layer problem, and at each layer it could target different, possibly conflicting, system aspects. Many existing solutions have addressed adaptation in a "local" way by only considering one system aspect at one layer; in contrast CLAM fosters a comprehensive approach able to address different layers and aspects concurrently, reason on the dependencies and consequences among them, and identify global solutions.

Validation Set-up & Result	
Name	Cross-Layer Adaptation of SBAs
Synopsis	CLAM is a prototype that *(i)* analyzes the effects and consequences of an adaptation trigger for the whole service-based system and *(ii)* addresses the negative influences on the system through a gradual construction of adaptation strategies.
Authors	Zengin, A., Marconi, A. , Pistore, M., Baresi, L.
Research questions	Supporting adaptation of service-based applications Cross-layer Adaptation
Case study	Smart Management of Taxi Reservations from http://scube-casestudies.ws.dei.polimi.it/index.php/Smart_Taxi_Case_Study.
Scenario	This contribution use a scenario to illustrate the cross-layer adaptation problem addressed. A service-based system (SBS) is defined and is composed of an *application layer* where the service-based application takes place along with its on functional process properties such as key performance indicators (KPI), a *service layer*, which corresponds to the partner services of the process forming a platform for it, and finally the underlying *infrastructure layer* for the composite service (application) and the partner services.
Method	Prototype
Description	Adaptation is a cross- and multi-layer problem, and at each layer it could target different, possibly conflicting, system aspects. For example, when reorganizing the composition, one could privilege the application's price, its speed, or the compliance with some external regulations. Many existing solutions have addressed adaptation in a "local" way by only considering one system aspect at one layer; in contrast CLAM fosters a comprehensive approach able to address different layers and aspects concurrently, reason on the dependencies and consequences among them, and identify global solutions. These solutions must harmonize layers and system aspects, and provide an integrated adaptation plan based on local activities. CLAM relies on a comprehensive high-level model of the application and of the layers behind it. Each model element is associated with a set of analyzers to understand the problem, solvers, to identify possible solutions, and enactors, to apply them on the element. The coordinated operation of analyzers, solvers, and enactors is governed by predefined rules that identify the dependencies, and consequences, between the elements of the model and run the different tools. For each adaptation need, CLAM produces a tree of alternative adaptations, identifies the most convenient one, and applies it.
Goal	The goal was to validate of the approach, implementing a prototype and evaluating it on a service-based application for taxi reservation.
Set-up	Supporting architecture for CLAM is given in Figure 1. Given a system meta model, CLAM has three fundamentals to solve the cross-layer adaptation problem:

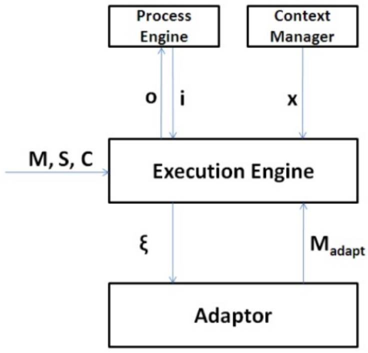
	 *Platform:* CLAM provides an integration platform where one can plug in analyzers and solvers and perform the cross-layer adaptation analysis. Through its pluggers the platform enables (i) to easily extend the system by including new artifacts to further improve the overall analysis, (ii) to thoroughly update the system with a different set of artifacts in order to target a totally different, new application domain. *Rules:* CLAM performs the entire adaptation analysis as a continuous execution of its predefined rules. These rules determine how to navigate the SBS meta model to identify the system elements affected by an adaptation, and subsequently how to invoke analyzers and solvers, and finally how to construct a cross-layer adaptation tree where the branches correspond to alternative adaptation strategies that can address the negative impacts of an initial adaptation. *Algorithm:* CLAM algorithm works based on the cross-layer system meta model, the artifacts connected to the platform, and the rules to coordinate these artifacts and perform the analysis upon receiving a new adaptation.
Inputs	CLAM needs as input the cross-layer representation of the application domain. This is done in terms of a *System meta model* that serves us in the following ways: (i) navigating the graph to reason on the overall analysis (ii) instantiating the model with concrete value assignments of system elements (we call them instant system configurations, e.g. M0, M1...) to keep track of alternative adaptation pro posals during the analysis. E.g., for our reference scenario, CPTS, we can instantiate a system configuration with the current BPEL file in use for the execution, partner services SMS, LS, PS that are provided by the current telecom provider Vodafone, and the TS that is provided by RadioTaxi Trento. Similarly we can instantiate each graph node with concrete system elements and overall graph instantiation will be an Mi.
Outputs	When a new adaptation is triggered by one of the solvers or a new adaptation need (problem) is triggered by one of the analyzers in the system, it comes to CLAM's rule engine. The engine initiates the collaborative impact analysis by executing the relevant rules one by one. Through this analysis, first affected system elements are identified in the meta model, the artifacts associated to these elements are selected, put in a queue, and invoked one by one. Each artifact makes its own local analysis and returns back a report to CLAM. Upon receiving reports CLAM triggers new rules and continues its operation until the system

	comes to a stable point where all the artifacts in the system are satisfied by some alternative global solutions. These solutions, kept in tree format, are the adaptation strategies that are a series of adaptation actions proposed by different solvers to meet the adaptation needs arisen by various analyzers.
Outcome	Positive
Experiences	The proposed solution relies on a comprehensive high level system model that allows to easily extend the framework by plugging-in new artifacts. We implemented the proposed framework and evaluate it on a service-based application for taxi reservation. In the future we plan to extend the approach with solutions to evaluate the identified adaptation strategies with respect to the past execution and adaptation history. Moreover, we will continue to evaluate the framework on application scenarios entailing higher level of complexity and requiring to deal with different system layers and concerns (e.g. Cloud-based applications).
References	Asli Zengin, Annapaola Marconi, Luciano Baresi and Marco Pistore. *CLAM: Managing Cross-layer Adaptation in Service-Based Systems.* SOCA 2011.
Glossary	Service-based Applications, adaptation, monitoring, adaptation strategies, Key Performance Indicators (KPIs).
Keywords	Cross-layer Adaptation, Service-based Systems

3.1.7.6 Adaptation of Service-based Business Processes by Context-Aware Replanning

In this work an adaptation approach is proposed. With it we are able to automatically adapt business processes to run-time context changes that impede achievement of a business goal. A formal framework that adopts planning techniques to automatically derive necessary adaptation activities on demand has been defined. The adaptation consists in identifying recovery activities that guarantee that the execution of a business process can be successfully resumed and, as a consequence, the business goals are achieved. The solution proposed is evaluated on a real-world scenario from the logistics domain.

Validation Set-up & Result	
Name	Adaptation of Service-based Business Processes by Context-Aware Replanning
Synopsis	We propose and evaluate a goal-based approach to business process adaptation in service-based applications. In our approach, adaptation activities are not explicitly represented. Instead, we build a formal framework that enables dynamic derivation of adaptation activities considering all aspects of the environment (current context, business policies, available services etc.) To support this, we 1) define a formal model for services and for service-based applications that enables adaptation. The evaluation concerns on the demonstration of our adaptation algorithm performance and the adaptation modelling overhead.
Authors	Antonio Bucchiarone, Marco Pistore, Heorhi Raik and Raman Kazhamiakin
Research questions	Context-aware SBA Adaptation
Case study	Car Logistic Scenario from [1].
Scenario	In order to illustrate the problems addressed in the paper and to evaluate our approach we use a case study from the logistics domain. The case study is inspired by the operation of the sea port of Bremen, Germany [1]. The port receives ships loaded with cars and has to organize the delivery of the cars to retailers. Before delivered, each car goes through a number of steps such as unloading, registration, storing, different types of treatment ordered by a retailer

	etc.
Method	Prototype
Description	In order to evaluate the effectiveness of the proposed approach, we have implemented the architecture that realizes the adaptation framework depicted in Figure 2.  Figure 2.Context-Aware Adaptation Architecture.
Goal	The goal of the evaluation was to study the complexity and overhead of the adaptation modelling, efficiency of the adaptation at run-time, and the scalability of the approach in a dynamic environment.
Set-up	Using our prototype implementation, we have conducted a series of experiments with the complex version of the logistics scenario. We have modelled and tested two variants of reference scenario: one to evaluate modelling overhead and the performance, and another to evaluate the scalability of the approach. The performance of the planning-based run-time adaptation has been tested on a 2GHz, 3Gb Dual Core machine running Windows.
Inputs	As input we have defined two variants the car logistics scenario. *Variant 1*. The reference business process is composed of 5 basic milestones (Ship Unloading, Storage, Technical Treatment, Consignment and Delivery) realized as an orchestration of 20 services. Besides that, 5 other services are also available for the adaptation activities. The adaptation logic of the application is based on 16 context aspects (e.g., car location, car status, storage ticket, etc.) and is governed by a set of 12 business policies over the process and involved services. In this scenario, 24 situations that require adaptation have been identified (e.g., light and severe car damages, occupied treatment stations, etc.). *Variant 2*. This version extends Variant 1 with a new service for in-place repairing of the cars (FastRepair). On the contrary, another repair service (SlowRepair) is dynamically removed from the repository. Note that in the scenario the process and the adaptation cases remain the same, while the way the process adapts to the same problem will change.
Outputs	Following our approach, adaptation modelling involved explicit representation of 16 relevant context properties (including their values, changes, and events), annotation of 25 services with corresponding context effects, and encoding of 12 business policies. Note, however, that the explicit modelling of 24 adaptation cases is not required by our framework. While, in general, such modelling effort may seem significant, in practice it is comparable to the effort needed to encode the adaptation logic using alternative methodologies like, for example, rule-based approaches. To verify this, we have analyzed the implementation of a very similar scenario following the approach presented in [2]. The adaptation was encoded as ECA (event-condition-action) rules defined using AGG tool. While the approach was different and the context properties and service annotations were not modelled explicitly, the effort necessary for the encoding of the rules and policies was comparable. It is important to notice that our modelling

	approach enables modularization and re-use of different specification elements. Indeed, the work on annotating services can be entrusted to service providers. If necessary, each service annotation can be reviewed and adjusted by its provider separately from other services of the application. On the contrary, in rule-based approaches any change in services and/or application policies may imply revision of the whole set of rules. Similarly, some elements of our model can be reused by other applications in a given domain: the context model and service annotations, once defined, may be adopted by different business processes. To evaluate *the performance* and the *outcomes of our adaptation* at run- time, we experimented with different adaptation cases of the above scenario. The results of the experiments unveil that, even though the vast majority of service protocols are fairly simple and have 2 to 4 BPEL-like activities, the constructed adaptation processes typically have more than 8 actions (in several cases, due to non-determinism, up to 60 actions) and quite complex structure. Nonetheless, the construction of the adaptation process never exceeded 4 seconds, which was much less than the normal execution time of the main process. This shows the high applicability of our algorithm to complex scenarios in different domains. To obtain an idea on the *scalability of our modelling approach*, we considered a second variant of the scenario, where the FastRepair service is introduced to replace the SlowRepair service. In our approach, this extension required only proper annotation of a new service. Moreover, such annotation can be completed by the service provider independently from the application, where the service is exploited. In the same situation, the rule-based implementation required modifying the set of adaptation rules related to the car damage and reverifying the whole rule system to avoid inconsistencies, which is much more complex and time consuming procedure.
Outcome	Positive
Experiences	Using our prototype implementation, we have conducted a series of experiments with the complex version of the logistics scenario. We have modelled and tested two variants of reference scenario: one to evaluate modelling overhead and the performance, and another to evaluate the scalability of the approach.
References	F. Bo¨se and J. Piotrowski. *Autonomously controlled storage management in vehicle logistics applications of RFID and mobile computing systems.* International Journal of RT Technologies: Research an Application, 1(1):57–76, 2009. H. Ehrig, C. Ermel, O. Runge, A. Bucchiarone, and P. Pelliccione. *Formal analysis and verification of self-healing systems*. In FASE, pages 139–153, 2010. Antonio Bucchiarone, Marco Pistore, Heorhi Raik and Raman Kazhamiakin. *Adaptation of Service-based Business Processes by Context-Aware Replanning*. SOCA 2011.
Glossary	Service-based application, Business Processes, adaptation strategies, Service Composition, Context, Planning.
Keywords	Service-based applications; adaptation; service composition; context.

3.1.7.7 Multi-layered Monitoring and Adaptation

Service-based applications have become more and more multi- layered in nature, as we tend to build software as a service on top of infrastructure as a service. Most existing SOA monitoring and adaptation techniques address layer-specific issues. These techniques, if used in isolation, cannot deal with real-world domains, where changes in one layer often affect other layers, and information from multiple layers is essential in truly understanding problems and in developing comprehensive solutions. We propose a framework that integrates layer specific monitoring and adaptation techniques, and enables multi-layered control loops in service-based systems. The proposed approach is evaluated on a medical imaging procedure for Computed Tomography (CT) Scans, an e- Health scenario characterized by strong dependencies between the soft- ware layer and infrastructural resources.

Validation Set-up & Result	
Name	Multi-layered Monitoring and Adaptation
Synopsis	In our prototype we have focused on the monitoring and adaptation of BPEL processes that are deployed onto a dynamic infrastructure.
Authors	Sam Guinea, Gabor Kecskemeti, Annapaola Marconi, and Branimir Wetzstein
Research questions	Multi-level and self-adaptation
Case study	The proposed approach is evaluated on a medical imaging procedure for Computed Tomography (CT) Scans, an e- Health scenario characterized by strong dependencies between the soft- ware layer and infrastructural resources.
Scenario	The application domain considered in this paper concerns the medical imaging procedure for Computed Tomography (CT) Scans. A CT Scan is an X-ray based Multi-layered Monitoring and Adaptation medical test that, exploiting sophisticated image processing algorithms, produces cross-sectional images of the inside of the body. These images can be further processed to obtain three dimensional views.
Method	Prototype
Description	A framework that integrates software and infrastructure specific monitoring and adaptation techniques, enabling multi-layered control loops in service-based systems has been proposed and evaluated. All the steps in the control loop acknowledge the multi-faceted nature of the system, ensuring that we always reason holistically, and adapt the system in a coordinated fashion. In the implemented prototype we have focused on the monitoring and adaptation of BPEL processes that are deployed onto a dynamic infrastructure. Building upon our past experiences we have integrated process and infrastructure level monitoring [1,2] with a correlation technique that makes use of complex event processing [3]. The correlated data, combined with machine-learning techniques, allow us to pinpoint where the problems lie in the multi-layered system, and where it would be more convenient to adapt [4,5]. We then build a complex adaptation strategy that may involve the software and/or the infrastructure layer [6], and enact it through appropriate effectors. The proposed approach is evaluated on a medical imaging procedure for Computed Tomography (CT) Scans, an e-Health scenario characterized by strong dependencies between the software layer and infrastructural resources.
Goal	The goal was to validate the proposed approach on a dynamic CT scan scenario.
Set-up	To establish self-adaptation, the framework proposed applies a slight variation of the well- known MAPE control loop. The integrated framework allows for the installation of multi- layered control loops in service-based systems. In the *Monitoring and Correlation step*, sensors deployed throughout the system capture run-time data about its software and infrastructural elements. The collected data are then aggregated and manipulated to produce higher-level correlated data under the form of general and domain-specific metrics. The main goal is to reveal correlations between what is being observed at the software and at the infrastructure layer to enable global system reasoning. In the *Analysis of Adaptation Needs* step, the framework uses the correlated data to identify anomalous situations, and to pinpoint and formalize where it needs to adapt. It may be sufficient to adapt at the software or at the infrastructure layer, or we may have to adapt at both. In the *Identification of multi-layer strategies* step, the framework is aware of the adaptation capabilities that exist within the system. It uses this knowledge to define a multi-layer adaptation strategy as a set of software and/or infrastructure adaptation actions to enact. A strategy determines both the order of these actions and the data they need to exchange to

	accomplish their goals. In the *Adaptation Enactment* step, different adaptation engines, both at the software and the infrastructure layer, enact their corresponding parts of the multi-layer strategy. Each engine typically contains a number of specific modules targeting different atomic adaptation capabilities.
Inputs	Using the reference case study, the CT Scan process is initially designed by a domain expert on the basis of the common medical procedure. The obtained process is a simple sequence of actions that does not embed any optimization with respect to its performance. The domain expert also specifies his goals for the quality of the medical procedure using a set of KPIs. An advanced user, such as a hospital IT technician, defines a set of adaptation actions that can be used to improve the process' performance. At run time we collect monitoring events both at the software and the infrastructure level and correlate them using EcoWare. Based on analysis, the adaptation need analyzer selects predefined adaptation actions which can improve the "overall process duration in case of whole body CT scans". This adaptation action is passed to the CLAM which updates the process model. In particular, the adapted process is handed over to DyBPEL, which manages the transition to the new process definition.
Outputs	An initial validation of the approach on a dynamic CT scan scenario has been done. We demonstrate how our approach can be used to automatically adapt this multi-layered system. In our future work we will continue to evaluate the approach through new application scenarios, and through the addition of new adaptation capabilities and adaptation enacting techniques.
Outcome	Positive
Experiences	We have presented an integrated approach for monitoring and adapting multi-layered service-based systems. The approach is based on a variant of the well-known MAPE control loops that are typical in autonomic systems. All the steps in the control loop acknowledge the multi-faceted nature of the system, ensuring that we always reason holistically, and adapt the system in a cross-layered and coordinated fashion. We have executed an initial validation of the approach on a dynamic CT scan scenario.
References	L. Baresi and S. Guinea. *Self-Supervising BPEL Processes*. IEEE Trans. Software Engineering, 37(2):247–263, 2011. A. Kertesz, G. Kecskemeti, and I. Brandic. *Autonomic SLA-Aware Service Virtualization for Distributed Systems*. In Proceedings of the 19th International Euromicro Conference on Parallel, Distributed and Network-based Processing, PDP, pages 503–510, 2011. L. Baresi, M. Caporuscio, C. Ghezzi, and S. Guinea. *Model-Driven Management of Services*. In Proceedings of the Eighth European Conference on Web Services, ECOWS, pages 147–154. IEEE Computer Society, 2010. R. Kazhamiakin, B. Wetzstein, D. Karastoyanova, M. Pistore, and F. Leymann. *Adaptation of Service-Based Applications Based on Process Quality Factor Analysis*. In ICSOC/ServiceWave Workshops, pages 395–404, 2010. B. Wetzstein, P. Leitner, F. Rosenberg, S. Dustdar, and F. Leymann. *Identifying Influential Factors of Business Process Performance using Dependency Analysis*. Enterprise IS, 5(1):79–98, 2011. A. Zengin, R. Kazhamiakin, and M. Pistore. *CLAM: Cross-layer Management of Adaptation Decisions for Service-Based Applications*. In Proceedings of the 9th International Conference on Web Services, ICWS, 2011.
Glossary	Service-based Applications, adaptation, monitoring, adaptation strategies, Key Performance Indicators (KPIs).
Keywords	Service-based applications, monitoring, adaptation

3.1.7.8 Service-based adaptation using models@runtime

Engineering software for dynamic adaptive service based environment cannot be achieved through traditional software engineering techniques as the computing infrastructure is highly dynamic, distributed, concurrent, heterogeneous, volatile and resources constrained. Thus a service-based application must inherently provide self-adaptation characteristics. In this work, we validate the possibility of using the models at runtime technique in dynamic service-based environment. This technique aims to ease the development of self-adaptable applications by driving runtime reconfigurations through the use of a global architecture model of the system. To use this technique, one of the main challenge is the ability to safely propagate reconfiguration policies, in order to ensure a consistent view on the architecture model on all computation nodes of the system. In S-cube, we propose a framework that relies on the models at runtime paradigm to maintain platform independent models of the current system architecture and its deployed configuration. We mainly focus on new algorithms that are able to propagate reconfiguration policies safely in order to preserve architecture models consistency between all computation nodes of the system. These algorithms use peer to peer techniques.

Validation Set-up & Result	
Name	Service-based adaptation using models@runtime
Synopsis	The approach allows the dissemination of adaptation policies on all the s-cube layers. This approach is validated by an implementation and an experiment. (www.kevoree.org)
Authors	Erwan Daubert, Grégory Nain, François Fouquet, Olivier Barais, Françoise André
Research questions	Cross-layer SBA adaptation Using models and aspect to design and adapt SBS Models@Runtime to check and optimize the adaptation plan
Case study	Crisis management system
Scenario	Context-driven adaptation based on requirements models and techniques
Method	Prototype
Description	The approach allows the dissemination of adaptation policies on all the s-cube layers. This approach is validated by an implementation and an experiment. (www.kevoree.org)
Goal	We have performed qualitative and quantitative evaluations of our approach, aiming at measuring the following indicators: (1) model propagation delay; (2) resilience to node infrastructure link failure; (3) ability to detect concurrent models and to handle reconciliation. For each indicator we have set up an experimental protocol, using case study metrics to simulate the system behavior on a grid in different configurations. We have chosen a large scale grid as an evaluation testbed. The use of a grid allows us to stress the algorithm by setting up a large number of nodes but it also brings us more control over the parameters of the experiment e. g. network failure simulations. In this way experiments are reproducible, and reproducibility is essential to our experimental protocol.
Set-up	Validation experiments share a common experimental protocol. Each experiment uses a set of logical node deployed on physical nodes within a computer grid. The logical nodes are instantiated in a separate Java Virtual Machines and use the reference Kevoree implementation for JavaSE (see http://www.kevoree.org). The experimental grid is an heterogeneous grid that contains nodes of mixed computational power and type. Each node is connected to a local area network at 100MB/s.
Inputs	Topology model. All our experiments take a bootstrap model as input, which describes the current abstract service-based architecture (i.e. in its platform independent form). This abstract model contains information on node representations, node logical links and node communication group instances and relationships. This node set and these relationships describe a topology of the system, which is used by our

	dissemination algorithm. In order to improve the simulation of our case study we use a random generator to create topology models that are organized in a cluster of clusters. **Global time axis traces.** In order to track the propagation of new configurations in this service-based system, we decorate the algorithm with a logger. This logger sends a trace for each internal state change (i.e new configuration or configuration reconciliation). In order to exploit temporal data on these traces without ensuring a global grid time synchronization we use a logger with a global time axis based on Java Greg Logger(http://code.google.com/p/greg/). More precisely, this type of logger is based on a client-server architecture. The server clock manages the global time reference. All clients periodically synchronize with the server, allowing it to store client latencies by taking into account clocks shift and network time transfer observed. Traces are emitted asynchronously by the client to the server, which then makes time reconciliation by adding the last value of latency observed for this client. All traces emitted by the server are therefore all time stamped accurately with the clock of the server. Finally, traces are chained by an algorithm to meet the following heuristic: a trace follows another one if it is the first occurrence which contains in its vector clock the originator node with it precise version number. Thus the final result for each experiment is a linked trace list on which we can precisely compute temporal results.
Outputs	We provide three experiments to check if models@runtime can be used to reconfigure service-based system **Propagation delay versus network load** This first experiment aims at performing precise measurements of the capacity to disseminate model configurations. These measures will take care of temporal (e.g. propagation delay) and volumetric (e.g. byte occupation on the network) properties. *Experimental protocol* As described in the common protocol subsection, measurements are performed on a computer grid. The probes injected in the Java implementation collect propagation delay and network occupation. After a bootstrap step on a topology model, a node chosen randomly reconfigures its local model with a simple modification. In practice this reconfiguration step computes a new model, moving a component instance from one node to another chosen randomly. This new model is stored in the node, and the reconfiguration awaits propagation. This new configuration is tagged with the identification of reconfiguration originator. Figure (http://www.irisa.fr/triskell/perso_pro/obarais/uploads/Main/exp1topo.png) shows the topology model used for multi hop communication in the 66 nodes of this configuration. The experiment is driven by the following parameters: – delay before starting an active check of the peers update (model synchronization); – activation of sending of change notification messages. To evaluate the impact of the second parameter, the experiment is run twice. In the first run, notifications are not used and the active synchronization delay is set to 1000 ms. In the second run, notifications are used and active synchronization delay is 15 s. In both cases, a reconfiguration is triggered every 20 seconds and each reconfiguration run takes 240 seconds, resulting in 12 reconfiguration steps. *Analysis.* The observed per hop propagations delays are presented as a percentile distribution graph (http://www.irisa.fr/triskell/perso_pro/obarais/uploads/Main/exp1delay.png). The values displayed are the raw values of absolute time logged divided by the

minimum number of hops between the target and origina- tor of the reconfiguration (the minimum being computed using a Bellman-Ford algorithm). The traffic volume from protocol messages is shown in Figure http://www.irisa.fr/triskell/perso_pro/obarais/uploads/Main/exp1net.png in KBytes per node per reconfiguration; the volume does not include payload. Absolute values of network consumption depends highly of implementation.

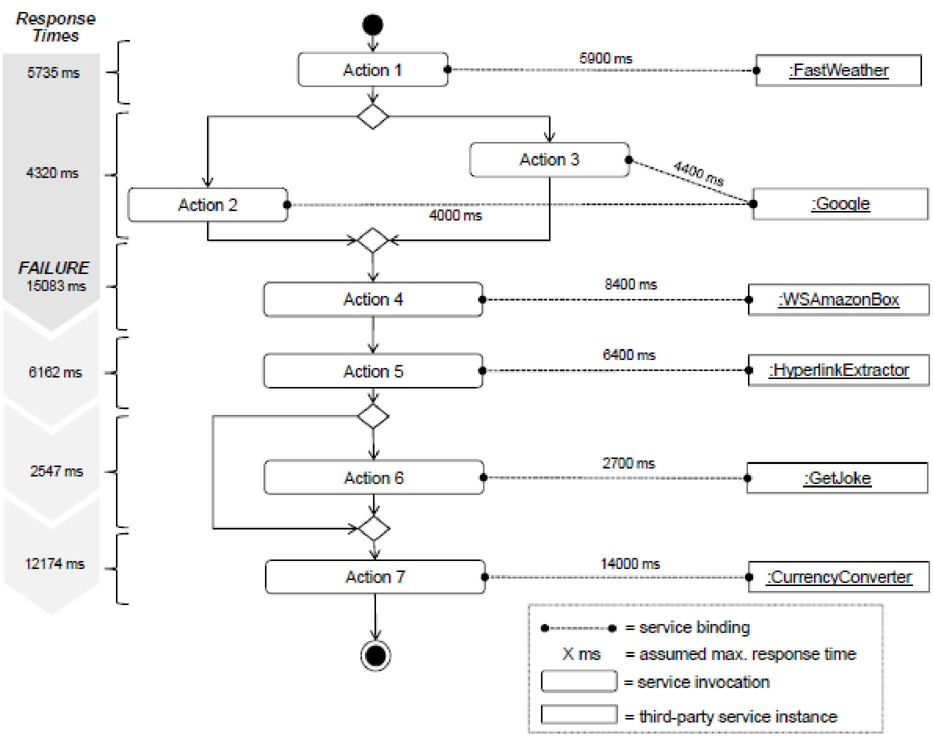
The use of notification reduces the propagations delay significantly: the average value decreases from 1510 ms/hop to 215 ms/hop. In addition, percentile distribution shows that the standard deviations of propagations are lower with in the version with notification. Thus this version of the dissemination algorithm has better scalability for large graph diameters.
However the gain on network usage is not as significant. Analysis shows that these results are affected by cycles in the topology. When using notification of change, nodes in cycles will create parallel branches configuration diffusion. This in turn will increase the number of conflict resolution to be done, and these resolutions increase network consumption unnecessarily, by exchanging the same model version. When notifications are not used, polling delays are large enough to avoid this concurrent configuration "flood". As the payload is a model with topology information, the notification algorithm could use this information to prevent flood.

Failures impact on propagation delay. The second experiment described below tests the ability of the algorithm to disseminate new models in a network with different failure rates.
Experiment protocol. The experiment protocol is similar to the first experiment's one. The topology model is enhanced to provide a mesh network with many different routes between nodes (see Figure http://www.irisa.fr/triskell/perso_pro/obarais/uploads/Main/exp2topo.png). At each run a modified model is pushed on a random node. The reconfiguration is similar to the previous experiment.
During each run, additional failures are simulated on links between two nodes, according to a Poisson distribution. The failure rate is increased at each run, thus the number of reachable nodes decreases. To perform this failure simulation we inject probes, which also monitor synchronisation events. At each run, the list of theoretically reachable nodes is computed and the initiator node waits for synchronisation events from theses nodes. When all events have been received we compute the average propagation delay. In short, this experiment aims at checking that every theoretically reachable node receives the new configuration.
Analysis. Figure (http://www.irisa.fr/triskell/perso_pro/obarais/uploads/Main/exp2.png) shows results of experiment #2. The histogram shows the rate of network failure for each run. The red curve displays the average propagation delay to reachable nodes (in milliseconds). Above a network failure of 85% the node originator of the reconfiguration is isolated from the network and therefore we stop the execution. With failure rate under 85% every node receives the new configuration and we can compute the propagation delay.

Concurrency reconfiguration reconciliation Our third experiment addresses the problem of reconciliation and conflict detection between concurrent model updates. This problem occurs often in the case study architecture because of the sporadic communication capabilities of our network of nodes. As a node can stay isolated for some time, reconfiguration data no longer reaches it. Furthermore, local reconfigurations can also occur in its subnetwork. Connection restoration may produce conflicting concurrent model updates. We rely on vector clocks to

	detect these conflicts and on the conflicting model updates themselves. Experiment #3 aims at checking the behavior of our algorithm in this conflicting updates situation. *Experiment protocol.* The experiment protocol is based on experiment #2. We use a similar grid architecture but with only 12 nodes. An initial reconfiguration is launched on the p00 node just after the bootstrap phase. All network links are up. Then a fault is simulated on the link between the node p00 and o00. Nodes o00, o01, o02 are then isolated. A new model is then pushed on node p00 and a different one on node o00. A delay of 1000 ms separates each reconfiguration and the algorithm is configured with a notification and a pooling period of 2000 ms. *Analysis.* Figure (http://www.irisa.fr/triskell/perso_pro/obarais/uploads/Main/exp3.png) shows results of experiment #3 derived from our branching algorithm traces. Three reconfigurations are represented as a succession of segments that show the propagation of updates. The first reconfiguration on the healthy network is represented in black (at time 0). Reconfiguration pushed on o00 (at time 2500) is represented in blue and the second reconfiguration pushed on p00 (time 2000) in red. The first reconfiguration propagates seamlessly to all nodes. At time 1500 a network failure is simulated. The second model given to p00 is propagated to all nodes except nodes reachable through o00 only. Similarly, the second model pushed on node o00 is not propagated to nodes after p00. At time 8000 we cancel the network failure simulated at time 1500. After a synchronization delay we can observe the branching of the two concurrent models as well as propagation of the merged version.
Outcome	positive
Experiences	We are confident that those techniques will become especially relevant in the setting of the "Internet of Services" and "Internet of Things", where applications will increasingly be composed from third party services, and run on top of smart devices which are not under the control of the service consumer and in which devices can be sometimes disconnect. This implies that applications and their constituent services need to be continuously adapted during their lifes. Models@runtimes seems to provide relevant mechanism and abstractions to support these feature. .
References	Brice Morin, Olivier Barais, Gregory Nain, Jean-Marc Jezequel. *Taming dynamically adaptive systems using models and aspects.* Proceedings of the 31st International Conference on Software Engineering (2009 Brice Morin, Olivier Barais, J-M Jézéquel, Franck Fleurey, Arnor Solberg. *Models@ runtime to support dynamic adaptation.* Computer Brice Morin, Thomas Ledoux, Mahmoud Ben Hassine, Franck Chauvel, Olivier Barais, Jean-Marc Jezéquel. *Unifying Runtime Adaptation and Design Evolution.* In IEEE 9th International Conference on Computer and Information Technology (CIT'09). Xiamen, China, Oct (2009) Francoise André, Erwan Daubert, Grégory Nain, Brice Morin, Olivier Barais. *F4Plan: An Approach to build Efficient Adaptation Plans.* In MobiQuitous. (2010) Grégory Nain, François Fouquet, Brice Morin, Olivier Barais, Jean-Marc Jézéquel. *Integrating IoT and IoS with a Component-Based approach.* In Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2010). Lille, France. (2010)
Glossary	Cross layer Adaptation, Adaptation Trigger , models@runtime, Adaptation, Workflow, Service Runtime
Keywords	Cross layer Adaptation, models@runtime, service runtime

3.1.7.9 Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification

Validation Set-up & Result	
Name	Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification
Synopsis	Based on assumptions about the SC's constituent services (derived from SLAs), the proposed approach SPADE formally verifies the SC against its requirements during run-time and adapts it if required.
Authors	Eric Schmieders and Andreas Metzger
Research questions	• Adaptation of QoS-aware Service Compositions based on Influential Factor Analysis and Prediction • Means to identify adaptation needs across layers • Associate adaptation strategies to the adaptation triggers • Perform runtime adaptation
Case study	None – Validation with abstract workflow
Scenario	The workflow used for validation is depicted in the figure below. The figure shows the workflow as a UML-based activity diagram. On the right hand side of the figure, the third-party services that are invoked by the SBA (i.e., by the actions of the workflow) are shown. On the left hand side, the service response times for one actual execution of the SBA are given, as one example set of response times of a walkthrough. 
Method	Prototype
Description	Our measurement of SPADE's efficiency is twofold. First, we examined unnecessary adaptations, i.e. false positives, as such adaptations could lead to avoidable costs, e.g., when replacing a free service with a commercial service to compensate for faults. Secondly, we count the amount of situations in which

<table>
<tr><td></td><td>SPADE cannot perform an adaptation. It can happen that a service invocation leads to a violation of an SLA, such that the end-to-end requirement is already violated. In those situations, the SBA instance obviously cannot be adapted preventively in order to avert that requirements violation, as the requirement has already been violated. Both values are expected to be low, as a low value implies a high number of cases where SPADE was successfully applied.</td></tr>
<tr><td>Goal</td><td>Experimental validations of comprehensive approaches like SPADE must be accurate and comprehensive as well. To reduce this complexity, we focus on SPADE's ability in identifying adaptation needs in our set of experiments.</td></tr>
<tr><td>Set-up</td><td>The performed experiments are based on a simulated execution of the example SBA's and its services. This enables the reproducibility of the test results for the exact same input data, thus allowing other researchers to reproduces the performed tests. We simulate the example SBA, introduced in Section 3. Specifically, we simulate the execution of the workflow and retrieve the response times for its constituent services from a large set of monitoring data. This dataset comprises real monitoring data, which was crucial for our experimental design. By using realistic monitoring data, we show the applicability of SPADE in a realistic setting. The number of SBA instances that can be experimentally assessed is limited by the size of the used dataset. To each service invocation within an SBA instance we assign one single data point from the dataset of the respective service. In the example SBA, this allows for the execution of 5884 different SBA instances.</td></tr>
<tr><td>Inputs</td><td>As input we use the scenario depicted above, as well as a set of response times published in:
Cavallo, B.; Di Penta, M. & Canfora, G. An empirical comparison of methods to support QoS-aware service selection Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, ACM, 2010, 64-70</td></tr>
<tr><td>Outputs</td><td>The SPADE approach has been applied to 5884 SBA instances (cf. row (a) in Table 1). 629 of those SBA instances have been executed without any assumption violations (b). During the execution of the complementary 5255 SBA instances assumptions have been violated (c) those SBA instances, SPADE has identified 604 preventive adaptation triggers (d).
<table>
<tr><th>Description</th><th>%</th><th>SPADE</th><th></th></tr>
<tr><td>SBA Instances Executed (a)</td><td></td><td colspan="2">5884</td></tr>
<tr><td>SBA Instances without Assumption Violation (b)</td><td>b/a</td><td>629</td><td>10.7%</td></tr>
<tr><td>SBA Instances with Assumption Violation (c)</td><td>c/a</td><td>5255</td><td>89.3%</td></tr>
<tr><td>SBA Instances with Adaptations (d)</td><td>d/a</td><td>604</td><td>10.3%</td></tr>
<tr><td>False Positive Adaptations (e)</td><td>e/a</td><td>72</td><td>1.2%</td></tr>
</table>
Table 1. False Positives in Relation to Executed SBA Instances
<table>
<tr><th>Description</th><th>%</th><th>SPADE</th><th></th></tr>
<tr><td>Service Invocations in Executed SBA Instances (a)</td><td></td><td colspan="2">32362</td></tr>
<tr><td>Actual Service Invocations until Adaptation Trigger (b)</td><td>b/a</td><td>30784</td><td>95.1%</td></tr>
<tr><td>Assumption Viol., where Adaptation is not Possible (c)</td><td>c/b</td><td>825</td><td>2.5%</td></tr>
</table>
Table 2. Invocations leading to Performance Violation</td></tr>
<tr><td>Outcome</td><td>72 of the adaptation triggers were false positives (e) (see tables above). Thus, 1.2% of the workflow instances would have been unnecessarily adapted. With respect to the challenging time constraint of 55 seconds, we consider the percentage of false positives extremely promising, especially, as an unnecessary adaptation does not imply an SLA violation. Furthermore, the amount of situations in which SPADE cannot adapt is very low as well. As depicted in row (c) of Table 2, a total of 825 out of 28624 service invocations, i.e. 2.5%, lead to situations where an adaptation is not possible. Nevertheless, this could still mean</td></tr>
</table>

	a threat in an emergency setting and has to be further scrutinized.
Experiences	As SPADE does not rely on historical data, SPADE overcomes some of the shortcomings of the existing solutions. The applicability of SPADE has been supported by experimental results, using monitoring data of real services. We plan to continue our work on preventive adaptation in two directions. First, we will combine SPADE with our PROSA approach. The PROSA approach is capable of predicting quality violations of individual services. The combined approach is expected to act in situations in which SPADE is not able to prevent requirements violations as intended. Secondly, we plan to apply SPADE in a cross-layer adaptation setting. In this setting, SPADE is expected to exploit the adaptation mechanisms of two different layers: the service composition and the service infrastructure layer. We expect that harmonizing the adaptation on both layers will increase the number of situations in which SPADE is able to compensate for deviations, which thus may increase SPADE's success in avoiding requirements violations.
References	Schmieders, E. & Metzger, *A. Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification*. ServiceWave, 2011
Glossary	Preventive Adaptation, Adaptation Trigger, Assumption, Service Orchestration, Monitoring, Adaptation, Workflow, Service Runtime, Runtime Adaptation, Constraint Solver, Adaptation Strategy, Detecting
Keywords	Proactive Adaptation, Adaptation Trigger , Assumption

3.1.7.10 Service Network Analysis and Prediction Tool (SNAPT)

A prototype tool is presented for visualizing service networks based on a proposed meta-model and transforming these models into a draft set of collaborative business process models for Eclipse BPMN tool and IBM WebSphere Business Modeler tool. Also, service network models are analyzed based on a simple cost-revenue model and Value Network Analysis (VNA) proposed by Verna Allee.

Validation Set-up & Result	
Name	Service Network Analysis and Prediction Tool.
Synopsis	We present a prototype tool for constructing service network models based on a proposed service network meta-model which encompasses concepts from both IT and business perspective on service systems. The tool also supports a transformation process which maps a service network model to a set of draft collaborative business process models following the BPMN standard (can be imported to Eclipse BPMN Editor) and the free-form language supported by IBM WebSphere Business Modeler
Authors	George Stratakis, Pantelis Petridis, Christos Nikolaou, Mariana Karmazi, Konstantinos Tsikrikas
Research questions	End-to-end processes in Service Networks (JRA-2.1)
Case study	S-Cube Automotive Case Study
Scenario	The authors follow the description of the case study focusing on the focal actor (AutoInc) and the final service offered by the network ("cars").
Method	Prototyping
Description	For the Automotive Case study, the authors offer a meta-model and a methodology for visualizing the service network of the focal actor by focusing on the final service offered by the network; the "car". In this context, the model was drawn with the use of SNAPT and properly KPIs were added to each service offered or consumed in the network. Furthermore, the set of the draft collaborative business process models was extracted and evaluated against the proposed standards.

Goal	The intended goal is to visualize the service network as described by the service network meta-model and evaluate the collaborative business process models based on the proposed methodology.
Set-up	SNAPT was used for the visualization of the SN models and the business process models. In the case of business processes, two tools were utilized; the Eclipse BPMN editor for the business process models that follow the BPMN standard, and the IBM WebSphere Business Modeler. Also, the Vensim simulation tool was used for importing the simulation models as extracted from SNAPT.
Inputs	The business partners and the services that each offers to the network including concepts like costs and revenues.
Outputs	The results show that the meta-model is adequate enough for modeling the business perspective of the service networks. In the scope of IT perspective, the transformation methodology covers adequate the vision of transforming service network models to business process models. The
Outcome	Positive
Experiences	We believe that the SNAPT tool and the supported methodologies will promote the "Internet of Services" because SNAPT is a unique tool currently, and serves as a bridge between business analysts who focus on business performance and the IT developers who focus on the underlying business processes and back-end systems that realize service networks. SNAPT serves as a hub between widely acceptable tools for business process modeling, such as Eclipse and IBM WebSphere Modeler, and simulation tools like Vensim.
References	C. Nikolaou, P. Petridis, G. Stratakis, M. Karmazi, *Service Network Analysis and Prediction Tool*, in Paper to be submitted. Further Reading: P. Petridis, *Towards a universal Service Network-centric framework to design, implement and monitor Services in complex Service Ecosystems: The Service Network Analysis & Prediction Tool (SNAPT)*, Department of Computer Science, University of Crete, Heraklion, Master Thesis (Msc) 2010. G. Stratakis, *Analyzing Service Networksfrom different perspectivesusingthe Service Network Analysis & Prediction Tool (SNAPT)*, Department of Computer Science, University of Crete, Heraklion, Master Thesis 2011.
Glossary	Service Network
Keywords	Service networks, Service Network Modeling, Business Process Models

3.1.8 Validation Method: Experiments with prototypes

3.1.8.1 Online Testing to Support Proactive Adaptation

Validation Set-up & Result	
Name	Online Testing to Support Proactive Adaptation
Synopsis	(see description below)
Authors	Osama Sammodi, Andreas Metzger, Xavier Franch, Marc Oriol, Jordi Marco
Research questions	• Online Testing for Quality Prediction • Online QA approaches • Usage-based online testing for proactive adaptation
Case study	-
Scenario	Artificial abstract workflow is used in the experiment.
Method	Experiments with Prototypes
Description	The evaluation meant to quantify the effectiveness of our approach in terms of precision p of a prediction (and thus use the results for the comparison). Thus,

	we take into account the number of "false positive" (nFP), "true positive" (nTP), "false negative" (nFN), and "true negative" (nTN): • In the "false positive" case, failure prediction predicts a deviation from the expected quality although the service turns out to work as expected when invoked during the actual execution of the SBA. • In the opposite case, i.e., when it turned out that the adaptation was indeed needed, it is counted as a "true positive". • Likewise, the "false negative" case refers to the situation in which failure prediction was not able to predict a deviation although the service turns out to fail during the execution of the SBA. • In the "true negative" case, it turned out that the adaptation was in fact not required. Considering the above four cases, this leads us to the following overall computation of precision p as the proportion of true negatives and true positives over the total number of positives and negatives (normalized in the interval [0, 100]): $p = 100 * (nTP + nTN)/(nTP + nFP + nTN + nFN)$
Goal	The goal of the validation is to evaluate the effectiveness of our approach. It was designed with the aim to provide evidence to support the following hypothesis: *The complementary use of online testing & monitoring (our approach) improves the precision of failure prediction (i.e., the ability to correctly predict violations of the expected service quality) when compared to monitoring in isolation.*
Set-up	The experiment is based on: (1) the simulation of an example SBA and its associated services, together with (2) the prototypical implementation of the online test case selection, execution and prediction components. Specifically, we simulate the execution of the workflow of an example SBA (100 instances) and retrieve the response times for its constituent services (i.e., for the respective point in "simulation" time) from a large set of QoS data. The QoS data was collected by invoking real services every one hour for about four months, providing up to 2000 data points (simulation period) after removing significant outliers that have values above 20s. This data set serves two purposes: (1) it is exploited as monitoring data to predict the quality of future service invocations; (2) it is used to determine whether at the current point in time a failure occurs and thus is used to compute the precision.. In our experiment we focus on response time as the QoS property to predict. Thus, we define one performance test case for each of the services of the SBA, assuming a representative workload for each service. As we know the number of SBA instances that will be simulated to be 100, we computed the usage probabilities for the individual services and used those usage probabilities serve as an optimal estimate of a usage model.
Inputs	A large set of QoS data (e.g. response times) for real Web services (e.g., Google, Amazon). The QoS data was collected by invoking real services every one hour for about four months.
Outputs	The below tables show the results of our experiment. We use $pT\&M$ to refer to the precision of the prediction of our approach, and pM to refer to the precision of the prediction using monitoring only. The results for both "s1" and "s3" (two services used in our abstract workflow) show that in general both approaches perform better when the number of last observations is smaller. Furthermore, the results clearly indicate that the precision of failure prediction using the complementary use of testing and monitoring (as advocated by our framework) is better than the prediction using monitoring data only.

	EXPERIMENTAL RESULTS FOR SERVICE "S3"

Data points	Avg. $p_{T\&M}$	Avg. p_M	Difference	Improvement
last point	67%	55%	12%**	22%
last 5	57%	50%	7%**	14%
last 10	51%	47%	4%*	9%

EXPERIMENTAL RESULTS FOR SERVICE "S1"

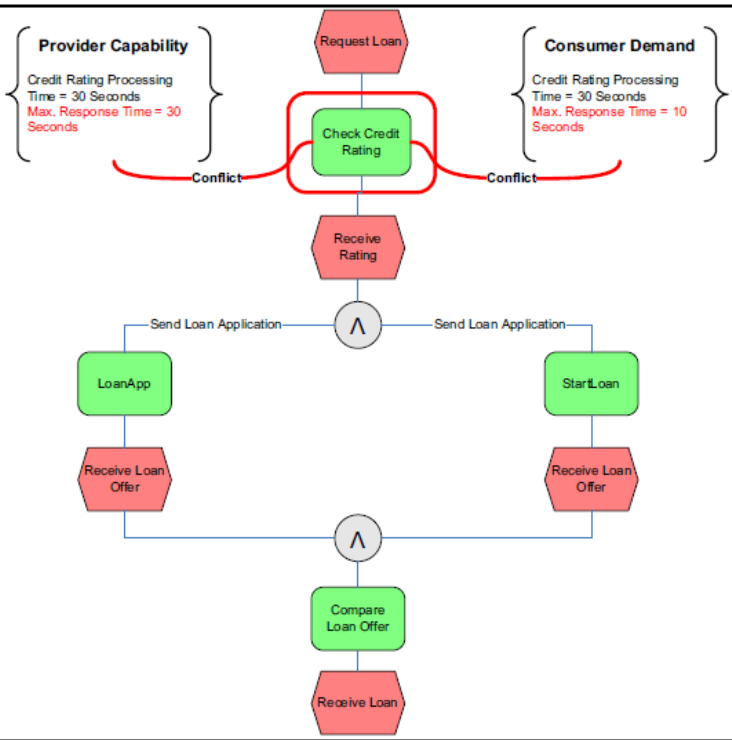
Data points	Avg. $p_{T\&M}$	Avg. p_M	Difference[3]	Improvement
last point	76%	73%	3%**	4%
last 5	72%	70%	2%	3%
last 10	73%	73%	0%	0%

Outcome	Positive
Experiences	Our approach is more beneficial when the number of monitoring data is small. For example, in the case of "s1", where the number of monitoring data is high, the improvement is only 4%. Whereas, in the case of "s3", where the number of monitoring data points is small, the improvement is higher (22%). Although, the maximum improvement is not high, the initial results are still promising. Considering the costs associated with our approach, the usage model can be used to estimate the number of monitoring data that will be available to decide ahead whether or not to perform online testing on specific services of an SBA. Still, the decision needs to be supported by more appropriate costs models which we will consider in future work.
References	Sammodi, O., Metzger, A., Franch, X., Oriol, M., Marco, J., & Pohl, K.. *Usage-based Online Testing for Proactive Adaptation of Service-based Applications*. In COMPSAC 2011 – The Computed World: Software Beyond the Digital Society. IEEE Computer Society. (2011) Metzger, A., *Towards Accurate Failure Prediction for the Proactive Adaptation of Service-oriented Systems*. (Invited Paper). In Proceedings Workshop on Assurances for Self-Adaptive Systems (ASAS), collocated with ESEC 2011. (2011)
Glossary	Analytical Quality Assurance, Failure, Failure Semantics, Fault, Monitoring, Proactive Adaptation, Quality Attribute, Quality of Service Characteristic, Quality of Service Constraint, Quality of Service Dimension, Quality of Service Level, Quality of Service-Based Adaptation, Reactive Adaptation, Service Fault, Service Level Agreement, Software Quality Assurance, Static Analysis, Testing, User Error, Validation, Verification, Quality of Service Negotiation, Service Level Agreement Negotiation, Level of Service
Keywords	SOA; online testing; usage models; operational profiles; monitoring; adaptation; quality prediction

3.1.9 Validation Method: Formal proof

3.1.9.1 Lifecycle Based Methodology for Negotiation Towards Service Level Agreements

Validation Set-up & Result	
Name	Lifecycle Based Methodology for Negotiation Towards Service Level Agreements.
Synopsis	We propose a stakeholder negotiation strategy for Service Level Agreements, which is based on prioritizing stakeholder concerns based on their frequency at each phase of the SBS development lifecycle. The first phase of this negotiation scheme is to identify the type and number of stakeholder roles involved in the SBS lifecycle. This information helped us to calculate their relevant frequencies at each phase, which in fact was used as a corresponding ratio of their participation in the negotiation process. Based on this information, an example conflicting situation was analysed using the potential number and types of the involved stakeholders. Finally, the identified stakeholder information was simulated to successfully predict the SLA cost associated with assigning priorities to both types of stakeholders; service providers and service consumers.
Authors	Hashmi, S.I., Haque R., Schmieders, E., Richardson I.
Research questions	Proactive SLA negotiation and agreement.
Case study	A service example from the following research paper: S. M. Huang, Y. T. Chu, S. H. Li, and D. C. Yen, “Enhancing Conflict Detecting Mechanism for Web Services Composition: A Business Process Flow Model Transformation Approach”, *Information and Software Technology*, vol. 50, no. 11, 2008. pp: 1069-1087.
Scenario	As part of our example scenario, we assume that a service consumer is negotiating with a service provider over the provision of an SBS (Figure 4). In addition, the figure shows the conflicted node using shaded rounded rectangle box. The conflicting situation triggers when the service consumer demands the system to check for the user’s credit rating in 5 seconds but the provider cannot make that available in less than 10 seconds. It is important to identify and negotiate these conflicts as they are likely to propagate among stakeholders, across different phases. For instance, the conflict between the service end-user and the service broker influences negatively the agreement between service broker and the credit rating service.

	
Method	Formal proof.
Description	We identify stakeholders and roles associated with them based on their key responsibilities at each phase of the SBS development lifecycle. Then, we identify how conflicts may occur between the service provider and the consumer while negotiating towards SLA. An example scenario is presented using the Collaxa BPEL *Loan Flow Service* to demonstrate the potential conflicts which can occur between the stakeholders.
Goal	Our goal has been twofold: propose a lifecycle based negotiation methodology which could involve stakeholders, information at each phase of the SBS development lifecycle, and validation of the approach in terms of numbers to demonstrate the usefulness of the approach.
Set-up	A quantitative stakeholder identification template was used to identify relevant stakeholders of the project, and the corresponding lifecycle phases where their activities resided. We mapped these stakeholders to the S-Cube lifecycle. Once stakeholders' types and count were identified, we distributed them into two different roles, such that: Total number of Stakeholder Roles (TR) = CR + PR Where CR = Σ (All stakeholder roles which are involved in consuming services, or in making use of them, e.g., application clients and end users.) PR = Σ (All stakeholders roles which are responsible for providing these services, e.g. application builders and service composers. services, e.g. application builders and service composers.) We used a *Loan Flow* service example to identify a scenario which may lead to conflicts on QoS between service provider and the service consumer. The identified conflicting node was mapped back to the SBS lifecycle to investigate the potential involvement of each lifecycle phases. Using this information, we measured the preference of a stakeholder's role on the conflicting node by calculating its relative frequency in comparison to the other roles. The greater the relative frequency value is, the more importance that stakeholder category has in the SLA negotiation process. These priority values were simulated to observe the potential impact of the corresponding stakeholders on the cost of the SLA negotiation.
Inputs	We have used stakeholder frequency information to prioritize and weigh the stakes of corresponding stakeholders in the SLA negotiation process. These relevant frequency values are considered as a priority mechanism in the

	negotiation process for that specific stakeholder role. These values are fed into our simulator.
Outputs	Using simulation cost equations, when priority is assigned to service provider roles, which is the likely case on the basis of higher relevant frequency values, the cost of SLA negotiation for the case is far less and stable as compared to service consumers'. Also, the graphical curve of the latter is quite abrupt and instable, which denotes that cost of assigning priority to service consumer roles will be far higher and inconsistent across different phases of the lifecycle. Using simulation cost equations, when priority is assigned to service consumer roles, which is the unlikely case on the basis of lower relevant frequency values, the maximum cost of SLA negotiation is even higher, with somehow similar abruptness and instability as identified in the previous case. But in comparison, the cost of SLA negotiation is still far less stable than the service provider roles.
Outcome	We have proposed a lifecycle based methodology for negotiation between service providers and service consumers. The results suggest that assigning priority based on the proposed approach could reduce cost of SLA negotiation.
Experiences	It is important to understand and implement a good negotiation process as it leads towards a formal agreement between the service provider and the service consumer in the form of an SLA. The proposed approach is likely to be applicable to a broader area with similar settings.
References	Hashmi, S.I., Haque R., Schmieders, E., Richardson I.,*Negotiation Towards Service Level Agreements: A Lifecycle Based Approach*, 7th IEEE World Congress on Services (SERVICES 2011), pp: 1-8.
Glossary	Service Level Agreement, Negotiation, Lifecycle.
Keywords	Service Based Systems (SBS), Stakeholders; Roles, Development Lifecycle, Service Level Agreement (SLA), Quality of Service (QoS), Loan Flow example.

3.1.10 Validation Method: Proof of concept

3.1.10.1 Managing evolving services

In an environment of constant change and variation driven by competition and innovation, a software service can rarely remain stable. Being able to manage and control the evolution of services is therefore an important goal for the Service-Oriented paradigm. We provided a unifying theoretical framework for controlling the evolution of services that deals with structural, behavioral and QoS level-induced service changes in a type-safe manner, ensuring correct versioning transitions so that previous clients can use a versioned service in a consistent manner. The table below reports the validation of this compatible evolution framework using the "automotive purchase order process scenario" to demonstrate its correctness and applicability. The application of the evolution framework to the scenario is found to be efficient, founded on solid theoretical foundations, and extensible to other technologies.

Validation Set-up & Result	
Name	Managing Evolving Services
Synopsis	Services are subject to constant change and variation, leading to continuous service redesign and improvement. However, service changes shouldn't be disruptive by requiring radical modifications in the very fabric of services or in the way that business is conducted.
Authors	Vasilios Andrikopoulos (Tilburg), Mike P. Papazoglou (Tilburg), Salima Benbernou (UCBL/ParisDescartes)
Research questions	Control of the evolution of services -- Under which conditions can services evolve without an impact to their context (their clients, services in the same service chain etc.)?

Case study	Automotive Case Study in: http://scube-casestudies.ws.dei.polimi.it/index.php/Automotive_Case_Study
Scenario	Purchase Order Processing Scenario: Placing purchase orders and schedule products delivery (IBM-SC-01)
Method	Proof of concept
Description	To control service development an application developer needs to know why a change was made, what are its implications and whether the change is complete. In a services environment, changes generally affect the service provider's application system. Typically service clients do not perceive the upgraded service immediately. Consequently, service-based applications may fail on the service client side due to changes carried out during a provider service upgrade. In order to manage changes in a meaningful and effective manner, the service clients using a specific service that needs to be upgraded must be considered when service changes are introduced at the service provider's side. Otherwise such changes will most certainly result in severe application disruption. Eliminating spurious results and inconsistencies that may occur due to uncontrolled changes is therefore a necessary condition for the ability of services to evolve gracefully, ensure service stability, and handle variability in their behavior. Thus, any service evolution management system has to be able to handle consistently and unambiguously the propagation, validation, and conformance of any kind of modifications applicable to a service. We proposed a formal causal model that addresses the structural effects of shallow (non-breaking for the clients) changes. In particular, we describe a versioning strategy to support multiple versions of services and a theoretical approach to deal with shallow changes in a such a way that service changes are analyzed, evaluated and constrained according to service change and versioning consistency and compatibility criteria. In this way, service-based applications and service clients are either expected not to be affected when a service changes and new versions are introduced, or they are expected to be able to cope with any service version that they encounter. For the purpose of validating our compatible evolution framework and providing a systematic way of looking at service evolution, we use the "Automotive Purchase Order Processing" case study developed jointly with IBM Almaden that the S-Cube Network of Excellence (www.s-cube-network.eu) uses as a validation scenario. The scenario is based on the Supply Chain Operations Reference model (SCOR) that provides abstract guidelines for building Supply Chains. This scenario is an example of how to realize SCOR level 3 activities using SOA-based processes for an enterprise in the automobile industry called Automobile Incorporation (aka AutoInc). AutoInc contains different business units, e.g. Sales, Logistics, Manufacturing, etc, and collaborates also with other partners like suppliers, banks, carriers, etc. In this scenario we assumed that the various process activities are implemented as services.
Goal	To demonstrate the correctness and applicability of the developed method.
Set-up	Use of the scenario as a demonstrator of the developed method. Within the context described above, we investigated on the effect of changes to different services. We developed various evolutionary scenarios, describing the required actions and proposed modifications to purchase order-processing services. We identified these scenarios either as maintenance actions (for example, optimizing the performance of some services) or as reengineering efforts (completely redesigning service interfaces).
Inputs	Evolution framework and scenario description
Outputs	Confirmation of the theory through the use of the method
Outcome	Positive.

Experiences	We defined the concept of T-shaped changes, i.e. changes that respect the compatibility of a service, and we demonstrated how we can reason on them using the case study. This approach was shown to be more efficient than the existing approaches on compatible evolution, founded on a solid theoretical foundation and more importantly, extensible to other technologies.
References	Michael P. Papazoglou, V. Andrikopoulos, and S. Benbernou. *Managing Evolving Services*, IEEE Software, May/June 2011, Vol. 28 No. 3, pp. 49-55. V. Andrikopoulos, S. Benbernou, and Michael P. Papazoglou. *On The Evolution of Services*, IEEE Transactions on Software Engineering. PrePrint, 01 Mar. 2011, IEEE Computer Society Digital Library, ISSN: 0098-5589.
Glossary	Lifecycle model, Service Based Application Construction, Service Composition, Evolution
Keywords	Service Evolution, Services Engineering, Versioning, Service Compatibility

3.1.11 Summary of the achieved IRF Building Blocks Validation

The validation experiments that are reported in the previous sections show a good distribution concerning the exploited experiment types and the coverage of the research challenges that S-Cube has expressed as part of its Integrated Research Framework (IRF).

The distribution of experiments through the categories defined in Section 2.1 is shown in Table 3. As it is natural in information technology, most of the experiments have concerned the development of prototypes. The researchers considerably exploited case studies as well. The only validation approach that has not been exploited is field study that, however, has been exploited in year 3 of the project.

Table 3. Classification of experiments with respect to the adopted validation method.

Validation method	No. of contributions
Controlled experiment	2
Case study	7
Field study	0
Action research	1
Survey	1
Literature review	1
Prototyping	10
Experiments with prototypes	1
Formal proof	1
Proof of concept	1
Total	**25**

Table 4 shows the mapping between the validation experiments and the research challenges. In particular, rows correspond to challenges and columns to the used validation methods. The numbers in the table correspond to the sections where the validation experiments are presented. This mapping has been performed by analysing the description of each experiment and the questions to which it aims to answer. When appropriate, we have allowed an experiment to map on more than one research challenge.

Almost all experiments have been associated to at least one research challenge, in some cases to more than one. Two experiments (presented in Sections 3.1.6.1 and 3.1.7.3) are marginal with respect to the S-Cube research challenges as they are focusing on specific techniques for process fragmentation, and therefore have not been added to the table. Furthermore, all challenges have been covered by at least one experiment, with the exception of the one on end-to-end quality definition language that has been closed in year 3 of the project.

Table 4. Mapping of validation experiments to the research challenges.

Research challenge	Controlled experiment	Case study	Field study	Action research	Survey	Literature review	Prototyping	Experiments with prototypes	Formal proof	Proof of concept
Definition of a coherent life cycle for adaptable and evolvable SBA									3.1.9.1	3.1.10.1
Measuring, controlling, evaluating and improving the life cycle and the related processes.		3.1.2.7								
HCI and context aspects in the development of service based applications				3.1.4.1						
Understand when an adaptation requirement should be selected							3.1.7.6, 3.1.7.9			
Identify best practices for SOA migration					3.1.5.1					
Comprehensive and integrated adaptation and monitoring principles, techniques, and methodologies							3.1.7.8			
Proactive Adaptation and Predictive Monitoring		3.1.2.4					3.1.7.4			
Context- and HCI-aware SBA monitoring and adaptation							3.1.7.6			
Mixed initiative SBA adaptation		3.1.2.2								
End-to-End Quality Reference Model										
Rich and Extensible Quality Definition Language							3.1.7.8			
Exploiting user and task models for automatic quality contract establishment							3.1.7.1			
Proactive SLA negotiation and agreement									3.1.9.1	
Run-time Quality Assurance Techniques	3.1.1.2							3.1.8.1		
Quality Prediction Techniques to Support Proactive Adaptation		3.1.2.4					3.1.7.4	3.1.8.1		

End-to-end processes in Service Networks		3.1.2.1, 3.1.2.5					3.1.7.10			
Business transactions in service networks		3.1.2.1								
Formal Models and Languages for QoS -Aware Service Compositions		3.1.2.3								
Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies		3.1.2.6					3.1.7.9			
Analysis and Prediction of Quality Characteristics of Service Compositions	3.1.1.2						3.1.7.9			
QoS Aware Adaptation of Service Compositions	3.1.1.2						3.1.7.9			
Multi-level and self-adaptation							3.1.7.5, 3.1.7.7			
Deployment and execution management							3.1.7.2			
Process mining for service discovery	3.1.1.1									

3.2 Validation of the IRF Integration

This year, the S-Cube researchers developed eight high level scenarios. These eight scenarios are described in the following subsections. Section 3.2.8 provides a summary of the aspects covered by the high level scenarios.

3.2.1 High Level Scenario: A Chemical Based Middleware for Workflow Instantiation and Execution

3.2.1.1 Detailed Scenario Description

In Service-Oriented Architectures (SOAs) Service Based Applications (SBA) are built by creating a business process that defines a concrete workflow composed of different partner services available via the network. Usually, the bindings between the business process and the partner services are predefined by statically referencing the corresponding service endpoints. When SBAs are required together with QoS specifications, and in open and highly changing environments (like the Internet of Things), this approach cannot be adopted since service availability, service quality, and user requirements may change in time according to market-driven demand-supply mechanisms. Given the dynamic nature of this reference scenario, the possibility of dynamically configuring and executing SBAs becomes a crucial requirement. The goal of the proposed chemical-based middleware presented in [1] is to address the problem of on-demand selecting and binding partner services in response to dynamic requirements and circumstances. The middleware architecture integrates late binding service selection mechanisms with a workflow execution module. The selection mechanisms are carried out at two levels. The first occurs at the composition request level and it allows selecting abstract services by taking into account global QoS requirements on the composition of services, such as the overall delivery time and the total price. Then, the second selection mechanism takes place at the concrete binding level. It allows further selecting suitable service implementation to forward the invocation request.

3.2.1.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	An SBA request is expressed as an *Abstract Business Process* (similar to a WS-BPEL process) represented by a set of *Abstract Services* (AS) structured as a directed acyclic graph according to their dependence constraints [2]. Each AS is a service interface referring to a group of implementations (*Concrete Services*) with different quality attribute values.	SBA Architect, Application Developer, Domain Expert	BPM, SCC
Construction	The *Workflow Instantiation* (WI) chemical-based component is responsible for building a possible *Instantiated Workflow* (IWF) by assigning each workflow activity to a suitable AS based on the global QoS constraints and the available CSs.	SBA Architect, Application Developer	SCC
Deployment & Provisioning	Check if workflow specification together with assumptions fulfill the SBA requirements [3], [5].	SBA Architect, Application Developer	SCC, SI

Operation & Management	A Monitoring module stores information on the quality of a specific CS after its invocation so to build historical information to drive adaptation.	Application Developer	SI
Identify Adaptation Needs	During workflow execution, an Evaluation Module checks whether the CS to be invoked can satisfy the user's requirement. If not, it deactivates the current selected CS is invoker and it selects another concrete service based on the historical invocation information.	SBA Architect, Service Consumer	SI
Identify Adaptation Strategy	Concrete Service substitution as adaptation strategy.	SBA Architect, Service Consumer	SI
Enact Adaptation	Adaptation strategies expressed as chemical rules that are triggered [4], [6] when adaptation needs occur.	SBA Architect, Service Consumer	SI

3.2.1.3 Achieved Integration

The high-level scenario enables service adaptability through late binding service selection mechanisms occurring at the service composition and service substitution at the service binding level. The scenario covers the phases of the SBA design/construction time and runtime. The high-level scenario shows a good integration of the research results and covers all phases of the lifecycle. The scenario focuses mostly on the Service Composition and Service Infrastructure layers (SCC, SI).

3.2.1.4 References

[1] C. Di Napoli, M. Giordano, J-L Pazat, C. Wang: *A Chemical Based Middleware for Workflow Instantiation and Execution*. ServiceWave 2010, 100-111.

[2] Di Napoli, C., Giordano, M., Németh, Z., Tonellotto, N.: *Using Chemical Reactions to Model Service Composition*. Proc. of the Second International Workshop on Self-Organizing Architectures (2010), 43-50

[3] Wang, C., Pazat, J.L.: *Using Chemical Metaphor to Express Workflow and Orchestration*. The 10th IEEE International Conference on Computer and Information Technology (2010).

[4] Banàtre, J.P., Fradet, P., Radenac, Y.: *Higher-Order Chemical Programming Style*. International workshop on unconventional programming paradigms 3566 (September 2004), 84-95.

[5] Z. Nemeth, C. Perez, and T. Priol, *Workflow enactment based on a chemical metaphor*, in Proceedings of the Third IEEE International Conference on Software Engineering and Formal Methods. Washington, DC, USA: IEEE Computer Society, 2005, 127–136.

3.2.2 High Level Scenario: Event-based Cross-Layer Monitoring and Adaptation for Semantic Web Services

3.2.2.1 Detailed Scenario Description

This high-level scenario is based on the approach presented in [11]. This paper describes a framework towards event-based cross-layer service monitoring and adaptation. This example demonstrates how this work, along with other S-cube approaches, can be applied on the S-Cube Service Lifecycle for the case of Semantic Web Services.

Initially, requirements are collected from the parties involved in order to determine the goals that need to be satisfied by the SBA. Then, based on these requirements, a SBA abstract workflow is designed. This workflow contains placeholders that are filled with actual Semantic Web services through a discovery process that takes into account both functional and non-functional characteristics. The conclusion of the discovery process results in a concretized workflow that implements the required goals of the SBA, as well as a specification of the resulting composite service, in terms of its inputs, outputs, preconditions and effects. The resulting process that implements the SBA is deployed and executed, with the execution being monitored. Monitoring spans across the three functional SBA layers and delivers events as soon as they occurred. Example of such events can be: various infrastructure failures, SLA violations, KPI violations etc. These events dictate the need for adaptation, which is accomplished by appropriate adaptation strategies, which may range from changes to the underlying infrastructure, over service substitution or SLA renegotiation, to business process modification. The adaptation strategy is defined after reasoning on a set of rules, which map patterns of monitored events to specific adaptation strategies. Finally, to enact adaptation the execution engine should be instrumented with a model repository supplying the appropriate information, such as service descriptions and requirements, layer dependencies, and metric and SLA models.

3.2.2.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	GRL proposed for expressing requirements in the form of goals [1] Semantic Web Services described using OWL-S [2], extended to support a solution to the frame problem [3]. Non-functional characteristics are expressed using OWL-Q [4].	SBA Architect, SBA Engineer, Domain Experts, Service Designer	BPM, SCC
Construction	Discovery process based on [5] Composition specification derivation realized by [6] SBA Workflow expressed in BPEL4SWS [7]	Service Provider, Composition Designer, Business Process Analyst, SBA Engineer	BPM, SCC
Deployment & Provisioning	SBA process deployed and monitored using Dynamo [8]	SBA Engineer, Application Developer	BPM, SCC, SI
Operation & Management	Monitoring SLA violations [9] Run-time Monitoring of Instances and Classes of Web Service Compositions [10]	Service Provider End User	BPM, SCC, SI

	Monitoring Infrastructure using Nagios [13]		
Identify Adaptation Needs	Monitored Event Patters detection [11]	Manager	BPM, SCC, SI
Identify Adaptation Strategy	Identify cross-layer adaptation strategies [11]	Manager	BPM, SCC, SI
Enact Adaptation	Aspect oriented techniques used to enact adaptation [12]	-	SCC

3.2.2.3 Achieved Integration

This high-level scenario enables cross-layer service monitoring and adaptation based on event and event patterns, focusing on Semantic Web services. It integrates approaches developed within S-Cube, as well as works developed independent of S-Cube, in order to present a unified approach that covers the entire S-cube lifecycle. The scenario takes all three SBA layers into consideration; however the approaches included are mainly related to the Service Composition and Coordination layer (SCC).

3.2.2.4 References

[1] GRL web site. http://www.cs.toronto.edu/km/GRL/

[2] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. *OWL-S: Semantic Markup for Web Services*. http://www.daml.org/services/owl-s/1.0/owl-s.pdf, 2004

[3] G. Baryannis and D. Plexousakis, *The Frame Problem in Web Service Specifications*. In PESOS '09: Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, pages 9-12, Washington, DC, USA, 2009, IEEE Computer Society.

[4] K. Kritikos and D. Plexousakis, *Requirements for QoS-Based Web Service Description and Discovery*. IEEE Transactions on Services Computing, Vol. 2, No. 4, pages 122-139, 2009.

[5] K. Kritikos and F. Patterno, *Service discovery supported by task models*. Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing System, pages 261-266, EICS 2010, Berlin, Germany, June 19-23, 2010.

[6] G. Baryannis, M. Carro and D. Plexousakis, *Towards Deriving Specifications for Composite Web Services*, Technical Report FORTH/ICS, Pending.

[7] J. Nitzsche, T. van Lessen, D. Karastoyanova and F. Leymann, *BPEL for Semantic Web Services (BPEL4SWS)*. OTM Workshops 2007, pages 179-188.

[8] L. Baresi and S. Guinea, *Dynamo: Dynamic Monitoring of WS-BPEL Processes*. In Proceedings of the Service-Oriented Computing - ICSOC 2005, Third International Conference, Amsterdam, The Netherlands, December 12-15, 2005.

[9] C Zeginis and D. Plexousakis, *Monitoring the QoS of Web Services using SLAs*, Technical.Report FORTH/ICS 404, May 2010 (http://www.ics.forth.gr/isl/publications/paperlink/technical_report_404_May_2010.pdf)

[10] F. Barbon, P. Traverso, M. Pistore, and M. Trainotti, *Run-time Monitoring of Instances and Classes of Web Service Compositions*. In: ICWS. pp. 63–71. IEEE (2006)

[11] C. Zeginis, K. Konsolaki, K. Kritikos, and D. Plexousakis, *ECMAF: An Event-Based Cross-Layer Service Monitoring and Adaptation Framework*, 5th Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC'11) co-located with ICSOC 2011
[12] D. Karastoyanova and F. Leymann, *Bpel'n'aspects: Adapting service orchestration logic*. In International Conference on Web Services (ICWS), 2009, pp. 222–229
[13] E. Imamagic and D. Dobrenic, *Grid Infrastructure Monitoring System Based On Nagios*. High Performance Distributed Computing, Proceedings of the 2007 Workshop on Grid Monitoring, June 2007.

3.2.3 High Level Scenario: Self-Optimizing Compositions in the Service Lifecycle

3.2.3.1 Detailed Scenario Description

This scenario is based on the self-autonomic system presented in [1,2,3,4,5]. Together, these papers outline the design of PREvent, a framework for autonomous prediction and prevention of SLA violations. To this end, the framework uses the seminal steps of the MAPE loop of autonomic computing (monitoring, analysis, planning and execution). These steps map to the phases of the S-Cube adaptation lifecycle as follows. Firstly, the service composition needs to be monitored, in order to gather the raw data necessary for deciding whether SLA violations are probable. This maps to the Operation & Management step. Secondly, SLA violations are predicted based on the collected runtime data. This maps to the Identify Adaptation Need step. Thirdly, a subset of the available runtime actions, which may be available to prevent SLA violations, need to be selected. This step implements the Identify Adaptation Strategy phase. Finally, the selected adaptation actions are executed. This implements the Enact Adaptation step.

3.2.3.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Operation & Management	SLA monitoring [1]	Service Provider, Service Consumer	SCC
Identify Adaptation Needs	Prediction of SLA Violations [2]	Service Provider	SCC
Identify Adaptation Strategy	Cost-Based Optimization of Service Compositions [3], Stepwise Optimization [4]	Service Provider	SCC
Enact Adaptation	Fragment-Based Adaptation of Service Compositions [5]	Service Provider	SCC

3.2.3.3 Achieved Integration

The high-level scenario enables a behaviour 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 lifecycle. All layers form service infrastructure (SI) layer, service composition and coordination layer (SCC), to business process modeling layer (SCC) are covered by the Scenario. Evidently, the scenario has a focus on the SCC layer.

This scenario enables self-managing service compositions, which adapt themselves when in danger of violating promised SLAs. The scenario covers only runtime aspects. However, the full runtime adaptation loop is covered by well-integrated S-Cube research results. The work from this scenario is complementary to other S-Cube research on design and development of adaptable service compositions. The work discussed in this scenario is geared mainly towards the service composition layer.

3.2.3.4 References

[1] P.Leitner, A.Michlmayr, F.Rosenberg and S.Dustdar: *Monitoring, Prediction and Prevention of SLA Violations in Composite Services*, presented at IEEE International Conference on Web Services (ICWS) Industry and Applications Track, 2010

[2] P.Leitner, B.Wetzstein, F.Rosenberg, A.Michlmayr, S.Dustdar and F.Leymann: *Runtime Prediction of Service Level Agreement Violations for Composite Services*, presented at 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing, co-located with ICSOC 2009, 2009

[3] P. Leitner, W. Hummer and S. Dustdar: *Cost-Based Optimization of Service Compositions*, IEEE Transactions on Services Computing (TSC). To appear.

[4] P.Leitner, W. Hummer, B.Satzger and S. Dustdar: *Stepwise and Asynchronous Runtime Optimization of Web Service Compositions*, presented at 12th International Conference on Web Information Systems Engineering (WISE) (short paper), 2011

[5] P.Leitner, B.Wetzstein, D.Karastoyanova, W.Hummer, S.Dustdar and F.Leymann: *Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution*, presented at International Conference on Service-Oriented Computing (ICSOC) , 2010

3.2.4 High-Level Scenario: Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification

3.2.4.1 Detailed Scenario Description

The high-level scenario is based on the approach published in [8]. The approach extends the work published in [1], which was reported in Deliverable IA-3.2.4 as integration scenario. This preliminary work was validated by a proof-of-concept implementation while the work in [8] was validated by a resilient validation towards the efficiency in detecting adaption needs. The validation uses real monitoring data in order to assess the behavior in a realistic setting. This validation strengthens the integration potential of the exploited S-Cube research by its results. Beside this, the approach was extended by an adaptation strategy component, thus covering the whole S-Cube lifecycle now. Both enhancements lead to the decision to report the evolved approach in this deliverable.

The reported work exploits several S-Cube techniques to identify menacing requirement violations based on monitored deviations. These techniques are combined and extended to an approach called SPADE. In this work, we demonstrate how the techniques for determining preventive adaptation work across the various phase of the lifecycle can be combined to avoid requirement violations.

In order to achieve this, SPADE equips SBAs with adaptation capabilities, empowering them to adapt themselves before a requirement is violated. To this end, SPADE uses run-time verification techniques, execution data of the monitored instances and assumptions concerning the SBAs' context, derived from Service Level Agreements (SLAs) of the constituent third-party services. Together, these mechanisms are used to predict the SBA performance, which enables the detection of menacing performance requirements violations of running SBAs. Being independent from past executions, SPADE can be used in settings where no models on computational costs or training data are available.

3.2.4.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Usage of the Quality Meta Model (QMM) proposed [5] Usage of ALBERT proposed to formalize Requirements [6]	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) **Fehler! Verweisquelle konnte nicht gefunden werden.** Formalization of Assumptions by using ALBERT [6]	SBA Architect, Application Developer	SCC
Deployment & Provisioning	Check if workflow specification together with assumptions fulfill the SBA requirements [3, 6]	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	SCC
Operation & Management	Monitoring during runtime [2]	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 [1,3]	-	SCC
Identify Adaptation Strategy	Create adaptation plan and exploit service substitution as adaptation mechanism of the strategy [4]	-	SCC
Enact Adaptation	Aspect orientated techniques used to enact adaptation [7]	-	SCC

3.2.4.3 Achieved Integration

The following achieved integrations are observable:

- The high-level scenario enables a preventive adaptation in order to permanently meet the SBA requirements.
- The scenario covers all phases of the design time and the runtime.
- By relating the research results, produced in S-Cube, to each phase, the high-level scenario shows their integration.

The high-level scenario shows a good integration of the research results and covers all phases of the lifecycle. That again, the service infrastructure (SI) layer is not covered. The scenario poses a strong focus on the service composition and coordination layer (SCC).

3.2.4.4 References

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

[2] 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.

[3] 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

[4] 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.

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

[6] 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.

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

[8] Schmieders, E. & Metzger, A. *Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification*. ServiceWave, 2011

3.2.5 High Level Scenario: Behavior Adaptation based on the S-Cube Service Lifecycle

3.2.5.1 Detailed Scenario Description

The high-level scenario is based on the approach presented in [1], and elaborated in [2],[3],[4], and [5]. This initial paper describes techniques to determine mis-behavior models in collaborative environments. In this work, we demonstrate how the techniques for determining proactive adaptation work together across the various phase of the lifecycle 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, the work in [1] identifies some general forms of mis-behavior in mass collaboration and sets the rules to handle the situation. After the collection of triggers and requirements the work in [2] describes an integration of the outlined VieCure Framework for behaviour adaptation in the field of SBA using the G2 testbed environment. While the work in [3] illustrates the technical details of the adaptation strategy in a programming model, both the works in [1] and [2] present the results and numbers on the simulated adaptation efforts. In the works of [4] and [5] the scenario was extended to include brokers and SLAs. Instead of continuing the adaptation idea in these works the focus was on mis-behavior prevention. As a central technique for prevention a LSP ranking scheme was introduced and evaluated under different assumptions in both works.

3.2.5.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Mis-behavior models are discussed in [1], as well as, design of the behavior adaptation FW.	SBA Architect, Service Deployer, Domain Expert,	SI, SCC
Construction	Programming Model [3] and infrastructure integration [2]	SBA Architect, Application Developer, Service Designer	SI, SCC
Deployment & Provisioning	Deployment scripts [2],[3]	SBA Architect, Application Developer, Business Process Administrator	SCC
Operation & Management	Adaptation policies [1], [2], SLA monitoring [3], ranking strategies [3], [4]	End User, Manager	SCC, BPM
Identify Adaptation Needs	Creation of adaptation polices and check against runtime situations [1], [2]	End User, Service Consumer	SCC, BPM
Identify Adaptation Strategy	Isolation and bypassing of failing nodes [1]. Trust estimation for alternative nodes [2]. Ranking of nodes according to monitored node properties [4], [5]	-	SCC
Enact Adaptation	Communication channel adaptation [1], alternative node selection [2], ranking [4], [5].	-	SCC

3.2.5.3 Achieved Integration

The high-level scenario enables a mis-behaviour 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 lifecycle. All layers form service infrastructure (SI) layer, service composition and coordination layer (SCC), to business process modeling layer (SCC) are covered by the Scenario. Evidently, the scenario has a focus on the SCC layer.

3.2.5.4 References

[1] Psaier, H.; Skopik, F.; Schall, D.; Dustdar, S., *Behavior Monitoring in Self-healing Service-oriented Systems*, COMSAC 2010, IEEE.

[2] Psaier, H.; Juszczyk, L.; Skopik, F.; Schall, D.; Dustdar, S., *Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems*, SASO 2010, IEEE.

[3] Psaier, H.; Skopik, F.; Schall, D.; Juszczyk, L.; Treiber, M.; Dustdar, S., *A Programming Model for Self-Adaptive Open Enterprise Systems*, MW4SOC 2010, ACM.

[4] Psaier, H.; Skopik, F.; Schall, D.; Dustdar, S., *Resource and Agreement Management in Dynamic Crowdcomputing Environments*, EDOC 2011, IEEE.

[5] Satzger, B.; Psaier, H.; Schall, D.; Dustdar, S., *Stimulating Skill Evolution in Market-Based Crowdsourcing*, BPM 2011, Springer.

3.2.6 High Level Scenario: Mining Lifecycle Event Logs for Enhancing Service-based Applications

3.2.6.1 Detailed Scenario Description

Service-Oriented Architectures (SOAs), and traditional enterprise systems in general, record a variety of events (e.g., messages being sent and received between service components) to proper log files, i.e., event logs. These files constitute a huge and valuable source of knowledge that may be extracted through data mining techniques. To this end, process mining is increasingly gaining interest across the SOA community. The goal of process mining is to build models without a priori knowledge, i.e., to discover structured process models derived from specific patterns that are present in actual traces of service executions recorded in event logs. However, we suggest to focus on detecting frequent sequential patterns, thus to consider process mining as a specific instance of the more general sequential pattern mining problem. Furthermore, we apply two sequential pattern mining algorithms to a real event log provided by the Vienna Runtime Environment for Service-oriented Computing, i.e., VRESCo The obtained results show that we are able to find services that are frequently invoked together within the same sequence. Such knowledge could be useful at design-time, when service-based application developers could be provided with service recommendation tools that are able to predict and thus to suggest next services that should be included in the current service composition.

3.2.6.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Usage of the “patterns” mined from the service logs to avoid failures within workflows of previously “invoked” services.	SBA Architect, Application Developer, Business Analyst, Business Process Analyst, Domain Expert, SBA Engineer	BPM, SCC
Construction	Usage of the “patterns” mined from the service logs within a “service” recommendation tool to build workflows of services.	SBA Architect, Application Developer	SCC
Deployment & Provisioning	Check if workflow specification together with assumptions fulfill the SBA requirements.	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	SCC
Operation & Management	Monitoring during runtime	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	-	SCC

Identify Adaptation Strategy	Service substitution as adaptation strategy	-	SCC
Enact Adaptation	Aspect orientated techniques used to enact adaptation	-	SCC

3.2.6.3 Achieved Integration

The high-level scenario enables the mining of previously used patterns of services to enable “service” recommendation to SBA Architects and Application Developers. The scenario covers the phases of the design time and runtime. The high-level scenario shows a good integration of the research results and covers all phases of the lifecycle. The scenario has a strong focus on the service composition and coordination layer (SCC).

3.2.6.4 References

[1] Anton Michlmayr, Florian Rosenberg, Philipp Leitner, Schahram Dustdar: *End-to-End Support for QoS-Aware Service Selection, Binding, and Mediation in VRESCo*. IEEE Transactions on Services Computing 3, 193-205 (2010).

[2] Baresi, L., Guinea, S., Pistore, M., Trainotti, M.: Dynamo + Astro: *An Integrated Approach for BPEL Monitoring*. In: Proceedings of the 2009 IEEE International Conference on Web Services (ICWS'09). pp. 230-237. IEEE Computer Society (2009).

[3] Leitner, P., Wetzstein, B., Rosenberg, F., Michlmayr, A., Dustdar, S., Leymann, F.: *Runtime Prediction of Service Level Agreement Violations for Composite Services*. In: Proceedings of the 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC'09). pp. 176-186 (2009).

[4] Michlmayr A., Rosenberg F., Leitner P., Dustdar S.: *Comprehensive QoS Monitoring of Web Services and Event-Based SLA Violation Detection*. In: Proceedings of the 4th International Workshop on Middleware for Service Oriented Computing (MWSOC'09). pp. 1-6. ACM, New York, NY, USA (2009).

[5] Mulo, E., Zdun, U., Dustdar, S.: *Monitoring Web Service Event Trails for Business Compliance*. In: IEEE International Conference on Service-Oriented Computing and Applications (SOCA). pp. 1-8 (2009).

[6] Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: *Service-Oriented Computing: State of the Art and Research Challenges*. IEEE Computer 40(11), pp. 38-45 (2007).

[7] Unger, T., Leymann, F., Leymann, F., Scheibler, T.: *Aggregation of Service Level Agreements in the Context of Business Processes*. In: Proceedings of the Twelfth IEEE Enterprise Distributed Object Conference (EDOC 2008). vol. 0, pp. 43-52.

[8] Wetzstein, B., Strauch, S., Leymann, F.: *Measuring Performance Metrics of WSBPEL Service Compositions*. In: Proceedings of the Fifth International Conference on Networking and Services (ICNS'09). IEEE Computer Society (2009).

3.2.7 High Level Scenario: Adaptation of Service-based Business Processes by Context-Aware Replanning

3.2.7.1 Detailed Scenario Description

Service-based business processes are typically used by organizations to achieve business goals through the coordinated execution of a set of activities implemented as services and service compositions. Since they are executed in dynamic, open and non-deterministic environments, business processes often need to be adapted to exogenous context changes and execution problems. In this paper we provide an adaptation approach that can automatically adapt business processes to run-time context changes that impede achievement of a business goal. We define a formal framework that adopts planning techniques to automatically derive necessary adaptation activities on demand. The adaptation consists in identifying recovery activities that guarantee that the execution of a business process can be successfully resumed and, as a consequence, the business goals are achieved. The solution proposed is evaluated on a real-world scenario from the logistics domain.

3.2.7.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	Goals Specification of abstract activities	Business Process Analysis, Domain Expert	BPM
Construction	Business Process Definition, Context Modeling, Services modeling with annotations	Business Process Analyst, Domain Expert, SBA Engineer	BPM, SCC
Deployment & Provisioning	Services and Business Processes Deployment, Context Model deployment	Business Process Analyst, SBA Modeler, Domain Expert	BPM, SCC
Operation & Management	Business Process Execution, Services Invocation, Context Properties checking, Context Evolution	End User, Business Process Analysist	BPM, SCC
Identify Adaptation Needs	Business Process Activities preconditions violations, Activities refinements.	-	BPM, SCC
Identify Adaptation Strategy	Adaptation plan generation using planning techniques	-	BPM, SCC
Enact Adaptation	Activity refinement, Business Process structural adaptation, Service selection and substitution	-	BPM, SCC

3.2.7.3 Achieved integration

This high-level scenario enables business process adaptations by using different mechanisms, from the simple activity refinement to the re-structuring of the overall process. All these mechanisms take into account the execution context and the set of available services. It exploits techniques developed within S-Cube and adopts planning techniques to automatically derive necessary adaptation activities on demand. The solution proposed is evaluated on a real-world scenario from the logistics domain and takes only two SBA layers into consideration the Business Process Modeling layer (BPM) and the Service Composition and Coordination layer (SCC).

3.2.7.4 References

[1] Antonio Bucchiarone, Marco Pistore, Heorhi Raik and Raman Kazhamiakin. *Adaptation of Service-based Business Processes by Context-Aware Replanning*. SOCA 2011.

[2] P. Bertoli, R. Kazhamiakin, M. Paolucci, M. Pistore, H. Raik, and M. Wagner. *Control Flow Requirements for Automated Service Composition*. In Proc. ICWS'09, pages 17–24, 2009.

[3] F. Bose and J. Piotrowski. *Autonomously controlled storage management in vehicle logistics applications of RFID and mobile computing systems*. International Journal of RT Technologies: Research an Application, 1(1):57–76, 2009.

[4] M. Pistore D. Shaparau and P. Traverso. *Contingent planning with goal preference*. In Proc. AAAI'06, pages 927–934, 2006.

3.2.8 High Level Scenario: Multi-layered Monitoring and Adaptation

3.2.8.1 Detailed Scenario Description

Service-based applications have become more and more multi-layered in nature, as we tend to build software as a service on top of infrastructure as a service. Most existing SOA monitoring and adaptation techniques address layer-specific issues. These techniques, if used in isolation, cannot deal with real-world domains, where changes in one layer often affect other layers, and information from multiple layers is essential in truly understanding problems and in developing comprehensive solutions. In this high-level scenario we present an integrated approach for monitoring and adapting multi-layered service-based systems. The approach is based on a variant of the well-known MAPE control loops that are typical in autonomic systems. All the steps in the control loop acknowledge the multi-faceted nature of the system, ensuring that we always reason holistically, and adapt the system in a cross-layered and coordinated fashion. We have validated it using a dynamic CT scan case study.

3.2.8.2 S-Cube Service Lifecycle Coverage

Service Lifecycle Phase	Argumentation that addresses the Lifecycle phase	Involved Stakeholder	Functional Layer
Requirements Engineering and Design	CLAM meta-model for multi-layer SBAs [1]. Specification of multi-layer KPIs.	SBA Architect, Application, Business Analyst, Domain Expert, SBA Engineer	BPM, SCC
Construction	Specification of WS-BPEL compositions. Definition of Cloud infrastructure [2] Derivation of monitoring rules from KPIs [3]	Application Developer	SCC,SI
Deployment & Provisioning	Deployment of Cloud Infrastructure [2] Deployment of monitoring captors [3]	SBA Architect, SBA Modeler, Application Developer, Business Process Administrator	SCC,SI
Operation & Management	Multi-layer monitoring during runtime [3]	End User	BPM, SCC,SI
Identify Adaptation Needs	Analysis of the SBA performances and identification of the adaptation need [4]	-	SCC,SI
Identify Adaptation Strategy	Identification of the multi-layer adaptation strategy [1]	-	SCC

Enact Adaptation	Multi-layer adaptation enactment exploiting layer-specific adaptation mechanisms (process adjustment, dynamic process substitution and deployment, Laysi for infrastructure).	-	SCC,SI

3.2.8.3 Achieved integration

The high-level scenario shows a good integration of the research results and covers all phases of the lifecycle. All layers form service infrastructure (SI) layer, service composition and coordination layer (SCC), to business process modeling layer (BPM) are covered by the Scenario.

3.2.8.4 References

[1] A. Zengin, A. Marconi, L. Baresi, and M. Pistore. *Clam: Managing cross-layer adaptation in service-based systems*. In 2011 IEEE International Conference on Service-Oriented Computing and Applications.

[2] A. Kertész, G. Kecskemeti, and I. Brandic. *Autonomic SLA-Aware Service Virtualization for Distributed Systems*. In Proceedings of the 19th International Euromicro Conference on Parallel, Distributed and Network-based Processing, PDP, pages 503–510, 2011.

[3] L. Baresi and S. Guinea. *Self-Supervising BPEL Processes*. IEEE Trans. Software Engineering, 37(2):247–263, 2011.

[4] R. Kazhamiakin, B. Wetzstein, D. Karastoyanova, M. Pistore, and F. Leymann. *Adaptation of Service-Based Applications Based on Process Quality Factor Anal- ysis*. In ICSOC/ServiceWave Workshops, pages 395–404, 2010.

3.2.9 Summary of the achieved IRF integration

In this deliverable we have reported eight high level scenarios that have been exploited to validate the integration of the IRF. All of them have considered and provided contributions to all phases of the S-Cube lifecycle (except one, which covers 4 phase) and, as such, they can be mapped on the first and, partially, on the second research challenge reported in Table 4. They are also contributing to the vertical elements of the IRF (i.e. JRA-1.1, JRA-1.2 and JRA-1.3) as they all concern with adaptation and quality assurance. Table 5 provides an overview of how they map on the three horizontal elements of the research framework, that is, Business Process Management (JRA-2.1), Service Composition and Coordination (JRA-2.2) and Service Infrastructure (JRA-2.3).

Table 5. High level scenarios and IRF horizontal elements.

IRF horizontal elements / high level scenarios	Business Process Management (BPM)	Service Composition and Coordination (SCC)	Service Infrastructure (SI)
Chemical-based middleware (3.2.1)	X	X	X
Mining lifecycle event logs (3.2.2)	X	X	
Adaptation of service-based business processes (3.2.3)	X	X	
Cross-layer monitoring and adaptation (3.2.4)	X	X	X
Self-optimizing compositions (3.2.5)		X	
Preventing performance violations (3.2.6)	X	X	
Behavior adaptation (3.2.7)	X	X	X
Multi-layered Monitoring and Adaptation (3.2.8)	X	X	X

Finally, Table 6 summarizes the types of stakeholders that are explicitly listed in the high level scenarios. As the focus of S-Cube is on Service-Based Applications (SBAs) more than on single service, stakeholders focusing specifically on the development of single services are not considered. Negotiation agents as well are not considered as negotiation was not a major concern of the work in the fourth year of the project.

Table 6. High level scenarios and involved stakeholders.

Type of stakeholder	High level scenarios
SBA Architect	3.2.1, 3.2.2, 3.2.4, 3.2.6, 3.2.7, 3.2.8
SBA Modeler	3.2.3, 3.2.6, 3.2.8
Application Developer	3.2.1, 3.2.2, 3.2.4, 3.2.6, 3.2.7, 3.2.8
Business Analyst	3.2.2, 3.2.3, 3.2.6, 3.2.8
Business Process Analyst	3.2.2, 3.2.4, 3.2.6
Business Process Administrator	3.2.6, 3.2.7, 3.2.8
Domain Expert	3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.6, 3.2.7, 3.2.8
SBA Engineer	3.2.2, 3.2.3, 3.2.4, 3.2.6, 3.2.8
Service Consumer	3.2.1, 3.2.5, 3.2.7
Service Architect	
Service Deployer	3.2.7
Manager	3.2.4, 3.2.7
Composition Designer	3.2.4
Service Designer	3.2.4, 3.2.7
Service Developer	
Negotiation Agent	
Service Provider	3.2.4, 3.2.5
End User	3.2.2, 3.2.3, 3.2.4, 3.2.6, 3.2.7, 3.2.8

4 Summarizing the achieved Y4 Validation and Integration

In this deliverable we present the validation experiments that have been performed both on the building blocks of S-Cube Integrated Research Framework (Section 3.1) and on the integration of all these parts (Section 3.2). The collected 25 validations show a complete coverage of the S-Cube research challenges that were active in the last year of the project. Moreover, the collected results include the usage of various validation methods. Among the ones identified in Section 2, only one has not been used in year Y4 but in the previous year.

The high level scenarios focus on the validation of the integration of the IRF by aggregating S-Cube research results to one IRF lifecycle walkthrough. The eight collected scenarios demonstrate a good integration of the S-Cube research results. Most of them cover both, the design and run time phases of the S-Cube IRF lifecycle. All three IRF horizontal elements (BPM, SCC and SI) are addressed. We observed a slight focus on the service composition and coordination layer (SCC), but this has been considered quite natural given the orientation of the S-Cube NoE.

Summarizing, the presented paper-based validations confirm the S-Cube IRF building blocks and, complementary to this, the high level scenarios demonstrate and strengthen the integration of the IRF. Thus, the validation results presented in this deliverable support the conclusion that the validation and the integration of the S-Cube research results have reached a high level of maturity.

References

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

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

[3] Tilburg (Ed): Initial Definition of Users Patterns and Methodologies. S-Cube Deliverable PO-IA-3.1.4, 2010.