Grant Agreement N° 215483

Title:	***Results of the Second Empirical Evaluation***
Authors:	*SZTAKI, TUW, UniDue, UniHH, UPM, CITY, CNR, FBK, INRIA, Tilburg, USTUTT*
Editors:	*Alexander Nowak (USTUTT), Eric Schmieders (UniDue)*
Reviewers:	*Zsolt Nemeth (SZTAKI)*
	Fabrizio Silvestri (CNR)
Identifier:	*PO-IA-3.2.7*
Type:	*Deliverable*
Version:	*1*
Date:	*09.03.2011*
Status:	*Final*
Class:	*External*

Management Summary

This deliverable is the continuation of deliverable PO-IA-3.2.6 and reports the results of empirical evaluation activities undertaken by S-Cube partners for the validation of research results in year three of the network. The systematic guidelines and templates used for the documentation of validation results and their related aspects are detailed. The set of collected validation results, documented according to these guidelines, is then presented. Finally, planned, upcoming evaluation activities are introduced.

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

Members of the S-Cube consortium:

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

Published S-Cube documents

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

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

The S-Cube Deliverable Series

Vision and Objectives of S-Cube

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

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

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

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

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

Table of Contents

List of Figures

List of Tables

1 Introduction and Workpackage Vision & Objectives

In this deliverable, we report the results of the empirical evaluation activities of the Integrated Research Framework (IRF) performed during the third year of the network. Previous results of the validation activities have been reported in deliverable PO-IA-3.2.6 [1], i.e. this deliverable is an update of deliverable PO-IA-3.2.6. The remainder of the deliverable is structured as follows: this section outlines the vision and the objectives of the workpackage WP-IA-3.2, including the relation to other workpackages and a general timeline of WP-IA-3.2. Section 2 describes the basic method which is used here. Subsequently, Section 3 applies the before-mentioned method to collect and document the validation results of the IRF elements developed in the third year of the network. Section 4 reports on future evaluation efforts and Section 5 finally summarizes important findings in this deliverable.

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

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

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

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

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

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

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

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

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

- Collection, definition, adaptation and extension of high-level validation scenarios along the service life-cycle. These scenarios will consider, for example,
  - selected application domains from the S-Cube case studies in WP-IA-2.2;
  - 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 life cycle view).
- Application of the selected scenarios.
- Evaluation of results as a basis for framework improvement.

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

1.3 Validation Object – The IRF

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

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

1.4 Relation with other Integration Workpackages

For the overall strategy in WP-IA-3.2 it is important to understand the inputs and outputs needed and, therefore, to understand the relations and dependencies with the other integration workpackages. These dependencies 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. [2]).

- *WP-IA-1.1 (Convergence Knowledge Model) – WP-IA-3.2*: The knowledge model provides the relevant glossary terms related to the validation results.
- *WP-JRA-1.1 (Engineering Principles, Techniques and Methodologies for Hybrid, Service-based Applications) – WP-IA-3.2*: In WP-IA-3.2 stakeholders are collected which are related to the high-level scenarios. The stakeholders are provided to JRA-1.1 in order to refine the stakeholders (collected in JRA-1.1).

1.5 Roadmap and Timeline of IA-3.2

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

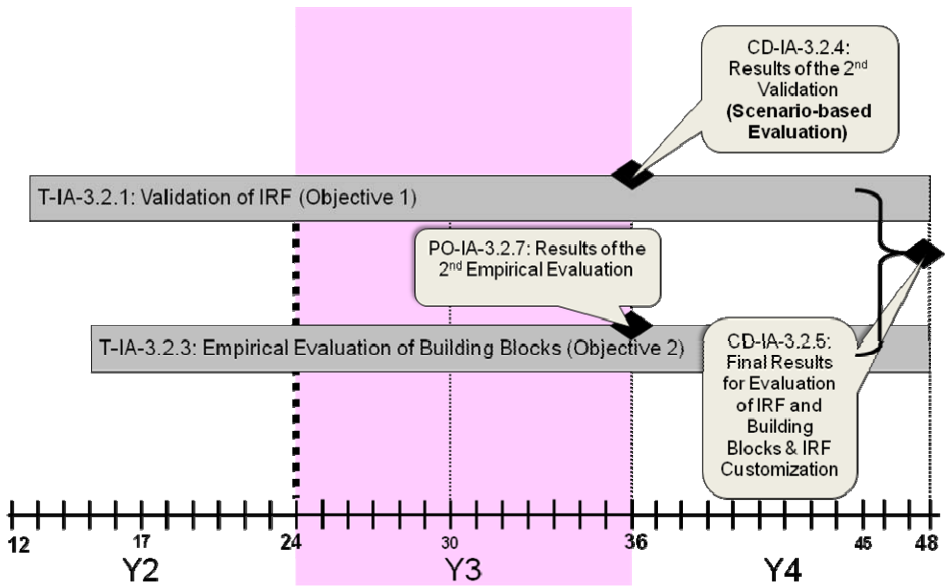


Figure 1 Timeline of WP-IA-3.2

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

2 Validation Strategy and Approach

At the end of the S-Cube project all major building blocks of the IRF should be evaluated and accessible for each partner in a structured way. Hence, within the work of this deliverable, we concentrate on the objective "*Empirical evaluation of the IRF building blocks*" (see Section 1.1). An empirical evaluation may focus on a single building block or may cross-cut multiple building blocks and should include different validation methods, e.g. a formal proof, a case study, an experiment, or a prototype in order to adequately validate the building blocks. This might also require experiment specific coupling of tools and infrastructures. Furthermore, for each IRF element produced by S-Cube its validation status should be known, e.g. it should be known which results were validated and in the positive case how strong this validation was. This aspect may be important for each partner when re-using the current S-Cube results. Since validation is a resource-intense activity, the objective of this deliverable is to share validation experiences and validation data, gained during the execution of the work within the network, between S-Cube partners in a structured way. Therefore we are using a predefined documentation template detailed in Section 2.1. The corresponding collection process performed is described in Section 2.2.

2.1 Structured Documentation Template

In order to properly validate the IRF building blocks, the evaluation results need to be documented in an uniform way. For this documentation we use the structural template depicted in Table 1, which is a minor extension to the structure proposed in [3]. This allows us to document the setup of the empirical evaluation, to analyze the results of the evaluation activities, and to provide a structured assess of the evaluation results. For consistency and better comparison of all validation results developed within the S-Cube project we decided to use the same template as in deliverable PO-IA-3.2.6 [1], except the research results, as they are no longer reported in the IRF.

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.
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 (cf. [5], p. 292 and [6]):

- *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.
- *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.

2.2 Collection Process

The documentation of evaluation results was performed following the agreed guidelines. The structural template capturing relevant aspects of the evaluation exercise such as the evaluation setup, the evaluation goals, specific procedures carried out, and outcomes were used (cf. Section 2). Each reported evaluation refers to at least one research question and challenge of the IRF. This ensures the evaluation of the different building blocks of the integration framework. Additionally, we allowed to prescribe own evaluation scenarios which exemplify how the evaluations are done with respect to real world examples. This differs from the previous deliverable PO-IA-3.2.6 where partners were restricted to use scenarios from those listed in PO-IA-3.2.1 ("Initial Definition of Validation Scenarios").

3 Validation Results

This chapter presents the set of the second empirical evaluations performed by S-Cube partners in year three of the network. It comprises 24 validations from eleven partners. The documented validation results, based on the introduced template, are presented in the following. In order to support a better usability of the results within WP-IA-3.1, the results are structured by their evaluation method. These include Prototyping, Experiments, Case Studies, Questionnaires, and formal proofs.

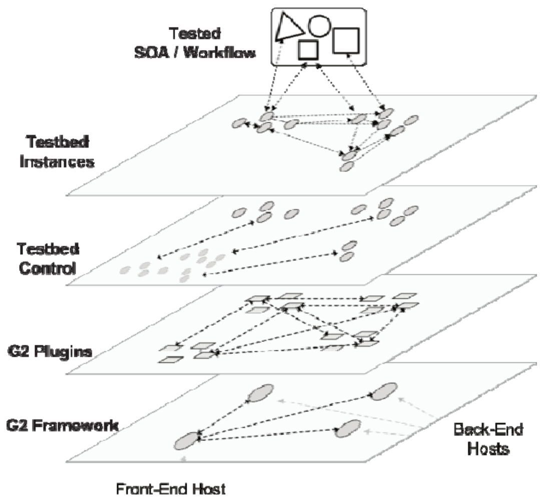
3.1 Validation Method: Prototype

3.1.1 Validation of script-based generation of dynamic testbeds

This section first tackles the problem of insufficient support for testing complex service-oriented systems. Table 2 documents the validation of a script-based generation of testbeds for evaluating the quality of complex services. Within a prototype the authors have comprised the emulation of Web services, clients, registries, bus systems, mediators, and other SOA components. The prototype, which has a positive validation outcome, allows to set up SOA testbeds and to manipulate their structure and behavior on-the-fly. Table 3 then documents the validation of a further prototype using the introduced approach. They used testbeds to emulate mobile worker's Web services and simulate QoS and dependability issues of mobile workers. Consequently, the prototype validates the test of workflows at runtime in realistic scenarios to get evidence about their correct execution.

Table 2 Script-based Generation of Dynamic Testbeds for SOA

Validation Set-up & Result	
Name	Script-based Generation of Dynamic Testbeds for SOA
Synopsis	The lack of support for testing complex service-oriented systems. The research community has developed various means for checking

	individual Web services but has not come up with satisfactory solutions for testing systems that operate in service-based environments and, therefore, need realistic testbeds for evaluating their quality.
Authors	Juszczyk, L.; Dustdar, S.
Research questions	How to create a SOA testbed that supports engineers in modeling testbeds and programming their behavior.
Scenario	-
Method	Prototype
Description	As complexity implies error-proneness as well as the need to understand how and where such complexity emerges, SOA-based systems must be tested intensively during the whole development process and, therefore, require realistic testbeds. These testbeds must comprise emulated Web services, clients, registries, bus systems, mediators, and other SOA components, to simulate real world scenarios.
Goal	We introduce the Genesis2 framework (Generating SOA Testbed Infrastructures, in short, G2) which allows to set up SOA testbeds and to manipulate their structure and behavior on-the-fly. It comprises a front-end from where testbeds are specified and a distributed back-end on which the generated testbed is hosted.
Set-up	G2 comprises a centralized front-end, from where testbeds are modeled and controlled, and a distributed back-end at which the models are transformed into real testbed instances. In a nutshell, the front-end maintains a virtual view on the testbed, allows engineers to manipulate it via scripts, and propagates changes to the back-end in order to adapt the running testbed. The G2 framework follows a modular approach and provides the functional grounding for composable plugins that implement generator functionality. The framework itself offers a) generic features for modeling and manipulating testbeds, b) extension points for plugins, c) inter-plugin communication among remote instances, and d) a runtime environment shared across the testbed.  Figure G2 Layer overview
Inputs	We use Groovy's Builder support which helps to create nested data

<table>
<tr><td></td><td>structures in an intuitive manner. The following sample demonstrates the convenience of builders:
<pre>
// hash map—based creation of web service model
def s1 = webservice.create("TestService")
s1.binding = "doc,lit"
s1.tags += "test"
def op = wsoperation.create("SayHello")
op.paramTypes += [name: String]
op.resultType = String
op.behavior = { return "hello $name" } // <— closure
s1.operations += op

// usage of model builder
def s2 = webservice.build {
  TestService(binding: "doc,lit", tags: ["test"]) {
    SayHello(name: String, result: String) {
      return "hello $name"
    }
  }
}[0]
</pre>
Listing G2 simple host script</td></tr>
<tr><td>Outputs</td><td>Realistic SOA Testbeds ready for emulation of scenarios and steering capabilities.</td></tr>
<tr><td>Outcome</td><td>Positive.</td></tr>
<tr><td>Experiences</td><td>-</td></tr>
<tr><td>References</td><td>Juszczyk, L., & Dustdar, S.: Script-based Generation of Dynamic Testbeds for SOA. In: Proceedings of the 2010 IEEE International Conference on Web Services (ICWS 2010). IEEE Computer Society (2010)</td></tr>
<tr><td>Glossary</td><td>Web Service, Performance, Planning, Monitoring Infrastructure</td></tr>
<tr><td>Keywords</td><td>Web-Service Testbeds, Emulation, Groovy</td></tr>
</table>

Table 3 Testbeds for Emulating Dependability Issues of Mobile Web Services

Validation Set-up & Result	
Name	Testbeds for Emulating Dependability Issues of Mobile Web Services
Synopsis	Today’s ubiquitous internet access has opened new opportunities for mobile workers. By using portable devices, the workers are not only able to access their company’s data and/or services from everywhere, but are also offering their own services for being accessible on-demand. The result is on the one hand a higher flexibility, in terms of coordination, but on the other hand poses various challenges to the company’s internal workflows due to the dynamic nature of mobility.
Authors	Juszczyk, L.; Dustdar, S.
Research questions	How can workflows be tested at runtime in realistic scenarios to get evidence about their correct execution?
Scenario	-
Method	Prototype
Description	For the purpose of emulating mobile worker’s Web services, we are using the G2 framework for generating the basic testbed. Furthermore, we apply plugins for simulating QoS and dependability issues of mobile workers.

Goal	Web services on portable devices suffer from two kinds of problems: unreliable connectivity, caused by the nature of wireless communication, and an unsteady availability of the human worker. If such services need to be incorporated into a company's workflow or service-oriented architecture, these systems must be able to handle the dynamics inherent in mobile computing and must be tested in simulated scenarios. Consequently, they require a testbed which emulates mobile workers and their dependability problems.
Set-up	G2 Testbed with emulated mobile workers: To achieve an effective emulation of these, we have extended G2 with two additional plugins: a network emulator, for low level fault injection, and a QoS plugin, for emulating QoS properties such as processing time, throughput, and scalability.

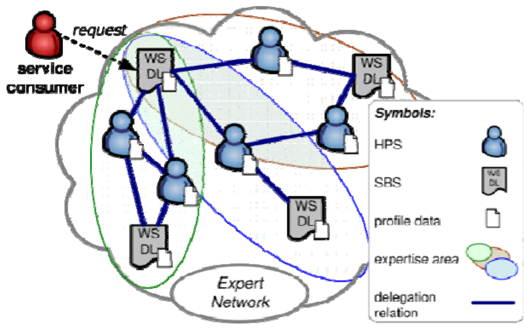
Inputs	Different profiles for changing network conditions, e.g.:
	beHos t1 { / / ma n i p u l a t e h o s t p r o p e r t i e s
	t c . l o s s = 0 .0 2 5 / / p a c k e t l o s s
	t c . d e l a y = 1500 / / p a c k e t d e l a y
	t c . c o r r u p t = 0 .0 0 1 / / p a c k e t c o r r u p t i o n
	}
	and for QoS properties, e.g.:
	s e r v i c e {
	/ / ma n i p u l a t e whole s e r v i c e ' s p r o p e r t i e s
	qos . p r o c e s s i n g t ime = 60 / / 1 minut e
	qos . t h r o u g h p u t = 5/60 / / 5 t a s k s p e r hour
	qos . a v a i l a b i l i t y = 0 . 9 8 / / 98% a v a i l a b i l i t y
	/ / ma n i p u l a t e QoS of s i n g l e WS o p e r a t i o n
	SayHe l lo . qos . t h r o u g h p u t = 10/60
	}

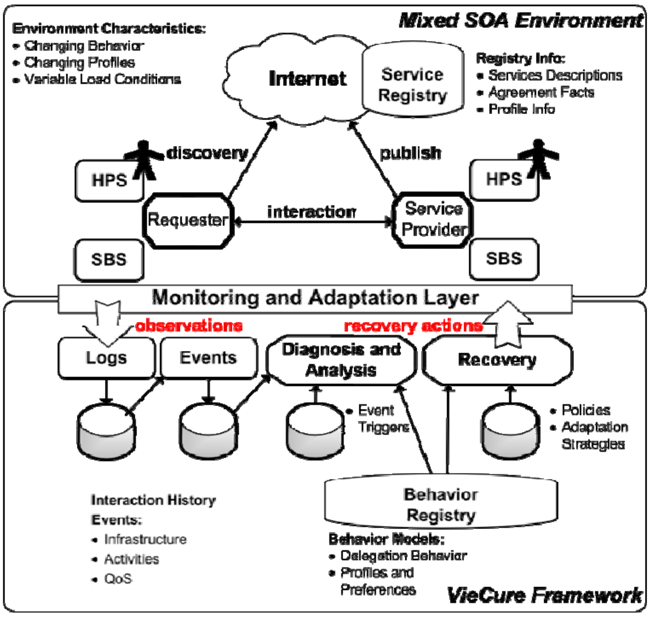
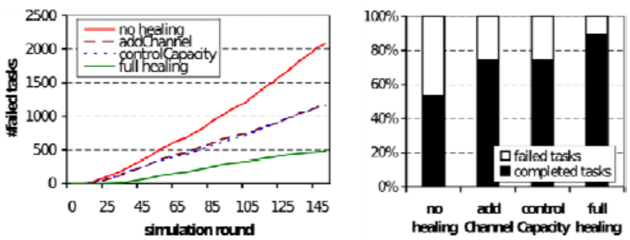
Outputs	Backend hosts and service emulating mobile devices with dynamically changing properties.
Outcome	Positive.
Experiences	-
References	Juszczyk, L., & Dustdar, S.: Testbeds for Emulating Dependability Issues of Mobile Web Services. In: 1st International Workshop on Engineering Mobile Service Oriented Systems (EMSOS). IEEE Computer Society (2010)
Glossary	Testing, Traceability, Dependability
Keywords	Mobile Web Services, Emulation, Testbeds

3.1.2 Validation of behaviour monitoring in service oriented architectures

Today, processes in collaborative environments are not restricted to single companies only, but may span multiple organizations, sites, and partners. Therefore, adaptations are necessary to keep the system fit and running (based on new/changing requirements) and to avoid misbehaving service environments. Table 4 documents a validation of a self-healing integration method of components in complex service systems. This approach is also focusing on mixed systems, i.e. it includes human provided services. The validation based on a simulation prototype shows a positive outcome. When adapting service systems it may occur that various circumstances cause inefficient task assignments in expert communities. Table 5 documents a validation of an approach to compensate misbehaviour of task delegation in service systems. The prototype that tested the effectiveness of adaptations with different task queue thresholds comes to a positive outcome.

Table 4 Behavior Monitoring in Self-healing Service-oriented Systems

Validation Set-up & Result	
Name	Behavior Monitoring in Self-healing Service-oriented Systems
Synopsis	-
Authors	Psaier, H.; Skopik, F.; Schall, D.; Dustdar, S.
Research questions	How can one apply self-healing paradigm to misbehaving service environments involving humans.
Scenario	 Today, processes in collaborative environments are not restricted to single companies only, but may span multiple organizations, sites, and partners. The actors perform assigned tasks with respect to prior negotiated agreements. Single task owners may consume services from external expert communities. For a single service consumer this scenario is shown in the Figure above. We model a mixed expert network consisting of Human-Provided Services (HPSs) and Software-Based Services (SBSs) that belong to different communities. The members of these communities are discovered based on their main expertise areas (depicted as shaded areas), and are connected through certain relations. A typical use case is the evaluation of experiment results and preparation of test reports in biology, physics, or computer science by third-party consultants (i.e., the Expert Network).
Method	Prototype, Evaluation
Description	Large-scale distributed applications become increasingly dynamic and complex. Adaptations are necessary to keep the system fit and running. New requirements and flexible component utilization call for updates and extensions. Thus, a challenge is the sound integration of new and/or redesign of established components. Integration must also consider changing dependencies. Today's SOAs are composed of loosely coupled services orchestrated to collaborate on various kinds of tasks. However, their benefit, modularity and an almost infinite number of combinations, fosters unpredictable behavior and as a consequence results in poor manageability.
Goal	To address the complexities, we introduce a self-healing approach enabling recovery mechanisms to avoid degraded or stalled systems. Extending the notion of self-healing by considering a mixture of human and service interactions observing their behavior patterns. Design and architecture of the VieCure framework supporting fundamental principles for autonomic self-healing strategies.

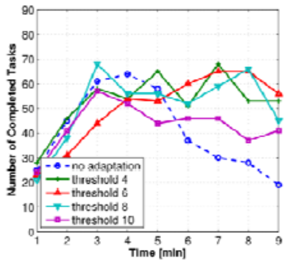
Set-up	The following figure shows the overall framework model comprising three main building blocks: SOA Environment consisting of human and software services, Monitoring and Adaptation Layer to observe and control the actual environment and the VieCure framework providing the main features to support self-healing actions. 
Inputs	In our experiments we evaluate the effectiveness of our recovery action algorithms designed for VieCure and simulated in a mixed SOA environment. For the complete algorithm please c.f. to the original work.
Outputs	The experiments measure the efficiency of a recovery action by the amount of failed tasks. An experiment consists of a total number of 150 rounds and a simulation environment with 128 nodes. During an experiment 4736 tasks are assigned to the nodes' network. In order to prevent an initial overload of a single node as a result of too many neighbor relations, we limited the amount of incoming delegations channels to a maximum of 6 incoming connections at startup. The resulting figures present on their left the total of failed tasks after a certain simulation round. The curves show the progress of different configurations of VieCure's diagnosis module. The figures on the right represent the ratio failed/processed tasks in percentages at the end of the experiments with an equal setting. 
Outcome	Positive.
Experiences	The evaluations in this work show that our elaborate recovery actions compensate satisfactorily the misbehaviors in a Mixed System (about

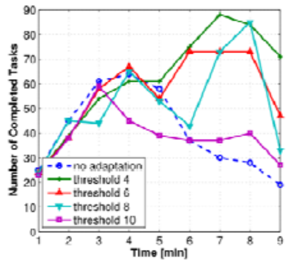
	30% higher success rate with equal distribution of behavior models). The success rates of the recovery actions depend on the environment settings. In all but one of the cases, deploying recovery actions supports the overloaded nodes resulting in a higher task processing rate. Important to note, that the failure rate increase near linearly even when recovery actions adjust the nodes' network structure. This observation emphasizes our attempt in implementing non-intrusive self-healing recovery strategies.
References	Psaier, H., Skopik, F., Schall, D., & Dustdar, S.: Behavior Monitoring in Self-healing Service-oriented Systems. In: Proceedings of the 35th Computer Software and Applications Conference. IEEE (2010)
Glossary	Self-Healing System, Adaptation Strategy
Keywords	Self-healing model, monitoring, recovery, mixed service-oriented system, delegation behavior

Table 5 Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems

Validation Set-up & Result	
Name	Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems
Synopsis	Many service-based applications demand for a mix of interactions between humans and Software-Based Services (SBS). Such applications are difficult to manage due to changing interaction patterns, behavior, and faults resulting from varying conditions in the environment.
Authors	Psaier, H.; Juszczyk, L.; Skopik, F.; Schall, D.; Dustdar, S.
Research questions	How can one Web-Service environment's interaction logs and trust metrics be utilized to equilibrate misbehaviour in the crowd.
Scenario	Crowdsourcing Scenario – Expert network
Method	Prototype, Evaluation
Description	Various circumstances may cause inefficient task assignments in expert communities. Performance degradations can be expected when a minority of distinguished experts become flooded with tasks while the majority remains idle. Load distribution problems can be compensated with delegations. Each expert in a community is connected to other experts that may potentially receive delegations. We assume that experts delegate work they are not able to perform because of missing mandatory skills or due to overload conditions. Delegation receivers can accept or reject task delegations. Community members usually have explicit incentives to accept tasks, such as collecting rewards for successfully performed work to increase their community standing (reputation). Delegations work well as long as there is some agreement on members' delegation behavior: How many tasks should be delegated to the same partner in a certain time frame? How many tasks can a community member accept without neglecting other work?
Goal	We identify two types of misbehavior: delegation factory and delegation sink. A delegation factory produces unusual (i.e.,

<table>
<tr><td></td><td>unhealthy) amounts of task delegations, leading to a performance degradation of the entire network. Work overloads lead to delays and, since tasks are blocked for a longer while, to a performance degradation from a global network point of view. A delegation sink can be characterized as a node that accepts more task delegations from neighbors as it is actually able to handle. Our approach provides a testing environment for such applications to address related challenges.</td></tr>
<tr><td>Set-up</td><td>The architecture presented in this paper follows the MAPE-K principle. A Logging Service provides Diagnosis and Analysis with timely status updates. Filtered status information populates the network model held by Diagnosis and Analysis module. During start-up the first interaction information is used to build the initial structure of the model. During runtime this information synchronizes the model with actual status changes observed on the network. Especially the interaction data filtered by the Behavior Monitor module allows Diagnosis and Analysis to draw conclusions from interactions about possible misbehavior at the services.</td></tr>
<tr><td>Inputs</td><td>In the experiments we simulate medium size teams of the crowdsourcing model. The teams comprise a total of 200 collaborators represented by Web services created by G2 Testbed scripts deployed to one backend instance. 20% of these members expose a delegation behavior the rest works on assigned tasks. We do not adapt from start. At start there is a period of 200 tasks with no adaptation. Then in an adaptation cycle the workers task queue size is monitored by tracing the delegation flow among the nodes. The difference between acknowledged assignments and complete or expired reported tasks results in the current task queue size at a particular worker.</td></tr>
<tr><td>Outputs</td><td>In our experiments we tested the effectiveness of adaptations with different task queue threshold triggers. The effectiveness is measured by the total task processing performance at the end of the experiment. Only completely processed and reported tasks went into the final result.





</td></tr>
<tr><td>Outcome</td><td>Positive.</td></tr>
<tr><td>Experiences</td><td>The final results show that the precise timing of multiple adaptations in a short term is most convenient for environment adaptation actions. However this has a trend to highly altering task processing results (e.g., approximately 40 task for a threshold 8 in the figure on the right). Comparing both, a strategy where the trigger matches the environments actor’s threshold of 6 is most practical in a balanced environment. Strategies with a threshold above 8 are infeasible for this</td></tr>
</table>

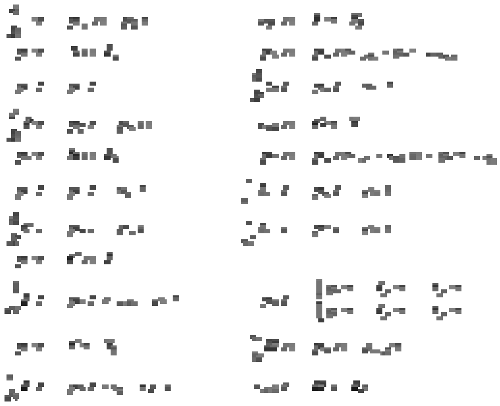
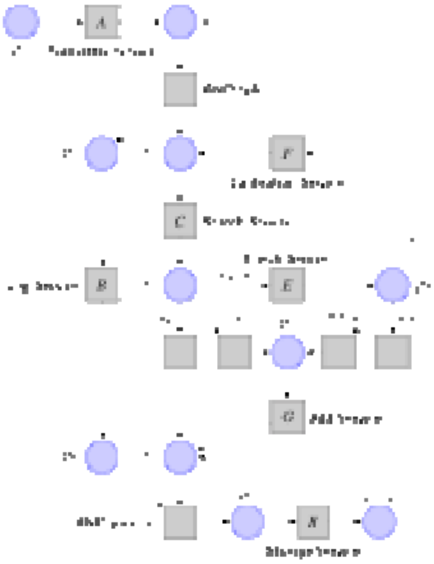
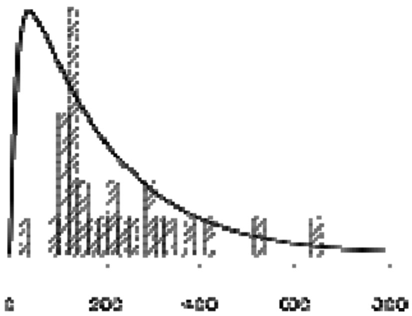
	setup. Generally the teleportation strategy performs better than mirroring, however requires the double and more adaptation actions.
References	Psaier, H., Juszczyk, L., Skopik, F., Schall, D., & Dustdar, S.: Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems. In: Proceedings of the Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 164–174. IEEE (2010)
Glossary	Adaptation Strategy, Automated Service Composition, Self-Adaptation
Keywords	Service-oriented collaboration, monitoring, self-adaptation, web service testbed

3.1.3 Validation of building dynamic systems of service orchestrations

Table 6 documents the validation of an approach focusing on dynamic models that are automatically generated out of orchestration descriptions. The positive outcome shows that the automatically derived dynamic model of a service orchestration corresponds to the analytically derived expected values from stochastic Petri-Net simulation.

Table 6 Building Dynamic Systems of Service Orchestrations with Provision Resources

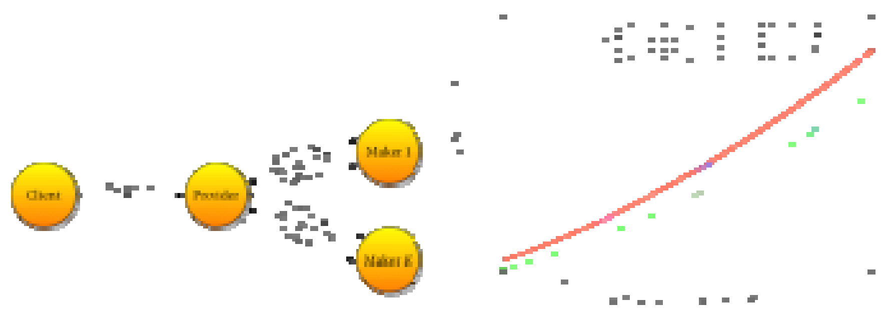
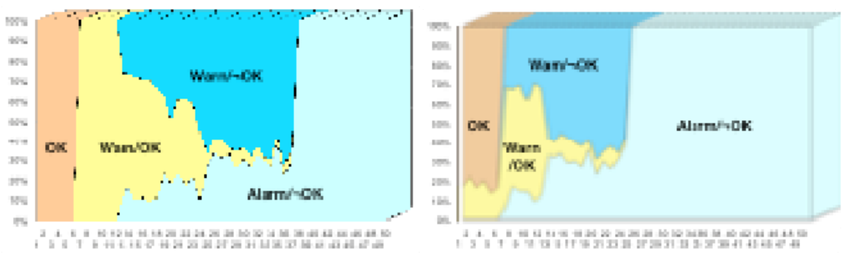
Validation Set-up & Result	
Name	Building Dynamic Systems of Service Orchestrations with Provision Resources
Synopsis	The validation result shows that the dynamic models automatically generated from orchestration descriptions correspond to the analytically derived expected values from stochastic Petri-Net simulation for median and the mean.
Authors	Ivanovic, D., Treiber, M., Carro, M., Dustdar, S.
Research questions	Dynamic modeling of service orchestrations and provision infrastructure
Scenario	A generic, service-independent service scenario, containing some long-running services.
Method	Dynamic system simulation in continuous time, compared with actual execution times. (Prototype)
Description	To establish whether the automatically derived dynamic (ordinary differential equations) model of a service orchestration, involving loops and parallel flows, we have compared the response of such dynamic system to a unit pulse (simulating arrival of a single request) with the empirical distribution of running times.
Goal	To show that the response of the system, given as the completed process outflow, and interpreted as a probability density curve over orchestration completion times, leads to descriptive parameters (namely mean and median) that correspond to the same parameters obtained from a set of measurements of running times from orchestration executions using dummy services
Set-up	Runtime environment, simulation environment.
Inputs	The simulation model and the orchestration with dummy component services.

	  Uniform, Gaussian, and Student-based jump distributions.
Outputs	Data sets from the empirical executions, and the numeric outflow from the dynamic model over execution times 
Outcome	Positive.
Experiences	Branching probabilities need to be well assessed to obtain reliable results from the simulation model.
References	Ivanović, D., Treiber, M., Carro, M., Dustdar, S.: Building Dynamic Models of Service Compositions With Simulation of Provision Resources. In: Proceedings of the 29th International Conference on Conceptual Modeling (ER 2010), Vancouver, Canada. Springer (2010)
Glossary	Service Orchestrations
Keywords	Dynamic Model, Modeling, Service Orchestration

3.1.4 Validation of data-aware monitoring, analysis and adaptation

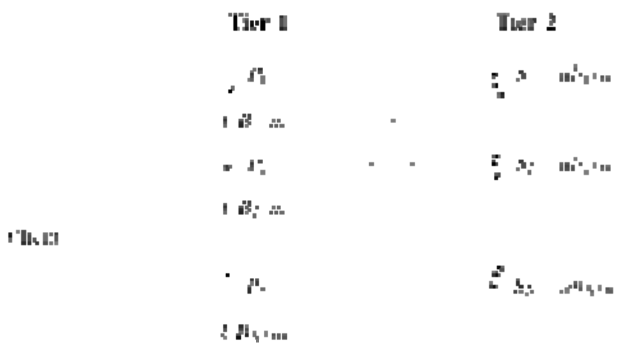
Table 7 describes an initial proposal for data-aware analysis of orchestrations in terms of validating the applicability of data-aware cost functions for expressing QoS properties. The simulation shows that a significant number of both Warnings (possible SLA violations) and Alarms (imminent SLA violations) correspond to the actual violations, while the number of false warnings is kept at a reasonable level. Table 8 then describes a simulation that investigates whether service matching based on cost functions over input data gives statistically significant improvements in the overall QoS (in this case: the execution time) of a service orchestration.

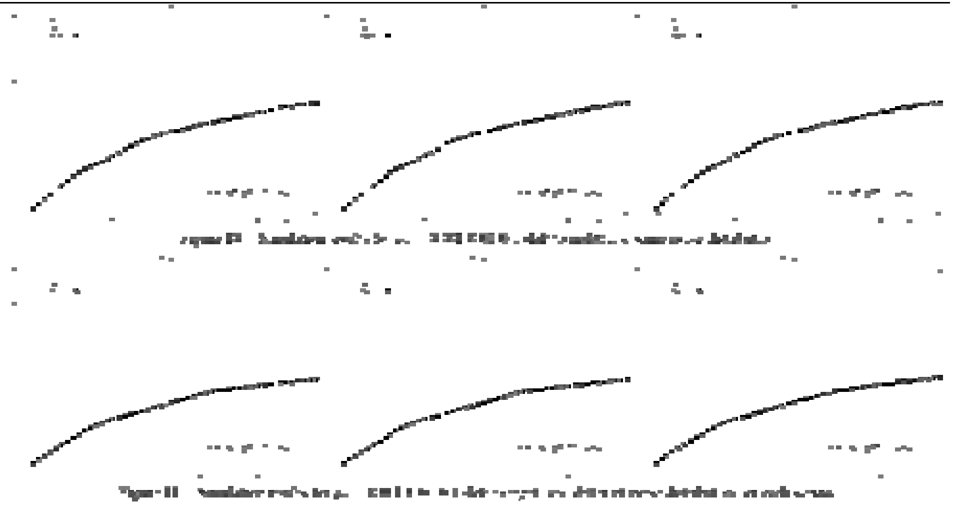
Table 7 An initial proposal for data-aware resource analysis of orchestrations with applications to predictive monitoring

Validation Set-up & Result	
Name	An initial proposal for data-aware resource analysis of orchestrations with applications to predictive monitoring.
Synopsis	Validation of applicability of data-aware cost functions for expressing QoS properties of service orchestration and provide for predictive monitoring.
Authors	Ivanovic, D., Carro, M., Hermenegildo, M.
Research questions	• Run-time Quality Assurance Techniques. • Quality Prediction Techniques to Support Proactive Adaptation.
Scenario	Automotive industry
Method	Simulation of a service network (Prototype)
Description	To establish whether QoS expressed with computational cost functions over input data can be effectively used for predictive monitoring.
Goal	Evaluate accuracy of the predictive approach.
Set-up	A single tier simulation setup, with the complexities of the backend services expressed as functions over input data: 
Inputs	Simulated execution time limit of 2500ms, and two simulation regimes: one with fault-free executions, and the other with a level of injected faults.
Outputs	The simulation outputs are given by the following graphs: 
Outcome	Positive.
Experiences	The simulation shows that a significant number of both Warnings (possible SLA violations) and Alarms (imminent SLA violations) correspond to the actual

	violations, while the number of false warnings is kept at a reasonable level.
References	Ivanovic, D., Carro, M., Hermenegildo, M.: An initial proposal for data-aware resource analysis of orchestrations with applications to predictive monitoring. In: Dan, A., Gittler, F., Toumani, F. (eds.), International Workshops, ICSOC/ServiceWave 2009, Revised Selected Papers, number 6275 in Lecture Notes in Computer Science, Vol. 6275. Springer (2010)
Glossary	Quality of Service, Data Sensitivity, Computation Cost
Keywords	Service Orchestrations, Resource Analysis, Data-Awareness, Monitoring

Table 8 Towards Data-Aware QoS-driven Adaptation for Service Orchestrations

Validation Set-up & Result	
Name	Towards Data-Aware QoS-driven Adaptation for Service Orchestrations
Synopsis	Validation of applicability of data-aware cost functions for expressing QoS properties of service orchestration.
Authors	Ivanovic, D., Carro, M., Hermenegildo, M.
Research questions	• Run-time Quality Assurance Techniques. • Quality Prediction Techniques to Support Proactive Adaptation.
Scenario	Automotive industry
Method	Simulation of a service network (Prototype)
Description	To establish whether service matching based on cost functions over input data gives statistically significant improvements in the overall QoS (in this case: the execution time) of a service orchestration.
Goal	Comparative advantage of data-sensitive vs. random and fixed preferences.
Set-up	 A two tier service network was simulated, with the actual computation costs in both tiers defined as a family of curves that depend on the input data: 
Inputs	The simulation was performed using two different noise distributions for execution times, and fault rate (noise level) of 0.001, 0.01 and 0.1.
Outputs	The simulation results are given in the following graphs:

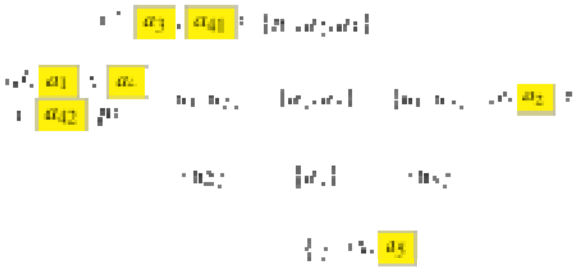
	
Outcome	Positive.
Experiences	The simulation results consistently show that the data-aware computational cost selection policy produces significantly better simulated QoS (execution time) than both the fixed preferences and the random approach, consistently across different noise levels.
References	Ivanovic, D., Carro, M., Hermenegildo, M.: Towards Data-Aware QoS-Driven Adaptation for Service Orchestrations. In: Proceedings of the 2010 IEEE International Conference on Web Services (ICWS 2010), Miami, FL, USA. IEEE (2010)
Glossary	Quality of Service, Data Sensitivity, Computation Cost
Keywords	Resource usage analysis; Data awareness; Monitoring; Adaptation.

3.1.5 Validation of interpreting sharing analysis results

Table 9 documents the validation of an approach that applies the sharing-based analysis to the problem of service composition fragmentation. The positive outcome of the prototype validates that sharing-based Independence-Driven Fragment Identification for Service Orchestrations can be achieved with this approach.

Table 9 Interpreting sharing analysis results to identify fragments

Validation Set-up & Result	
Name	Interpreting sharing analysis results to identify fragments.
Synopsis	Assigning different levels of attributes (such as access controls, reliability, precision, confidentiality, etc.) to input data, and representing how that data is manipulated and shared throughout the orchestration, can be used for determining relevant attributes of individual activities, and thus for assigning activities to fragment, based on interpretation of the sharing analysis results.
Authors	Ivanović, D., Carro, M., Hermenegildo, M.
Research questions	How to apply the sharing-based analysis to the problem of service composition fragmentation
Scenario	The health-care scenario.
Method	Applying automated tools for sharing analysis of logic program to an appropriate representation of the workflow, and interpreting the results of the analysis to construct the resulting lattice of fragment levels. (Prototype)
Description	Using “shadow” logic variables, different input data items to an orchestration can be organized into a (complete and finite) lattice of assigned levels. The sharing analysis applied to a representation of the workflow in the form of a logic program produces an abstract substitution that describes possible sharing settings

	between the variables that represent the input, the intermediate, and the resulting data items, as well as the activities themselves. By interpreting that resulting lattice using a minimal number of new "shadow" variables, a new lattice is constructed with all variables, which preserves ordering from the original lattice.
Goal	To establish correspondence between the initial and the final lattice.
Set-up	CiaoPP analysis suite for logic programs.
Inputs	Concrete substitution for the input data items / levels. 
Outputs	The interpretation of the resulting abstract substitution as the lattice of variables ordered per level.  (Highlighted variables denote activities)
Outcome	Positive.
Experiences	Explicit representation of the top element in the input lattice helps interpreting the resulting information.
References	Ivanović, D., Carro, M., Hermenegildo, M.: Automatic Fragment Identification in Workflows Based on Sharing Analysis. In: Weske, M., Yang, J., Maglio, P., Fantinato, M. (Eds.), Service-Oriented Computing – ICSOC 2010. Lecture Notes in Computer Science. Springer (2010)
Glossary	Fragmentation, Service Composition
Keywords	Fragment Identification, Sharing Analysis, Service Orchestration

3.1.6 Validation of the applicability of process migration to existing process description languages

For enabling a distributed execution of processes it is necessary to identify process fragments. As today many various process description languages exist, Table 10 shows a validation that the migration data meta-model [8] is also applicable to existing process description languages. The used prototype shows that the model is applicable to existing process engines and corresponding process description languages.

Table 10 Applicability of process migration to existing process description languages

Validation Set-up & Result	
Name	Applicability of process migration to existing process description languages
Synopsis	The validation shows that the migration data meta-model is applicable to existing process description languages.
Authors	Hamann, K., Zaplata, S.
Research questions	Context-Aware Execution of Distributed Processes
Scenario	none

Method	Prototype
Description	As an alternative to physical process fragmentation, the concept for realizing logical process fragmentation on the basis of process migration provides more flexibility by allowing for the distribution of running process instances at runtime while respecting the guidelines of the process modeler. The validation has also shown, that the proposed process migration data meta-model is applicable to at least two important process description languages (XPDL and WS-BPEL).
Goal	The goal of this validation is to show that the proposed migration data meta-model is applicable to existing process engines and corresponding process description languages, i.e. DEMAC with XPDL and Sliver with WS-BPEL.
Set-up	A prototype implementation has been applied to two existing process management systems: first to the DEMAC process engine which executes XPDL 1.0 processes and, second, to the Sliver process engine which executes a subset of WS-BPEL 2.0 processes. Both process engines can be applied also for mobile process management and had to be modified in order to implement the proposed management API.
Inputs	• DEMAC (Distributed Environment for Mobility-Aware Computing) platform • Sliver process engine • XML Process Definition Language (XPDL) specification • Business Process Execution Language for Web Services (WS-BPEL) specification
Outputs	The results from the detailed analysis of the process description languages XPDL and WS-BPEL are summarized in Table 26 and Table 27 (see appendix). The tables show how respective control flow constructs and other characteristics (such as privacy and flexibility) behave when adopting the process runtime migration model. The results are compared to the general concept of physical process fragmentation. In summary, it has shown that the process runtime migration data meta-model is applicable to the existing process description language XPDL and WS-BPEL. However, particular control flow constructs need a special attention and privacy can only be realized by artificially masking private process parts. Nevertheless, process migration allows for more flexibility in selecting the most suitable process engine at runtime while still allowing for respecting the interests of the process designer by determining specific participants or selection algorithms.
Outcome	positive
Experiences	The performed validation confirmed the advantage and applicability of the introduced approach and was therefore very reasonable.
References	Zaplata, S., Hamann, K., Kottke, K., Lamersdorf, W.: Flexible Execution of Distributed Business Processes based on Process Instance Migration. In: Journal of Systems Integration (JSI), 1(3). Prague (2010)
Glossary	Service Orchestration, Process Fragmentation, Migration, Runtime Process Migration
Keywords	DEMAC, Sliver, XPDL, WS-BPEL

3.1.7 Validation of cross-organizational process monitoring in service choreographies

Monitoring of processes across organizations requires that partners agree on monitoring events they provide to each other. Table 11 presents a validation which shows based on a prototype implementation and a purchase order processing scenario how monitoring agreements can be realized based on a WS-BPEL, BPEL4Chor, and WS-* technologies.

Table 11 Cross-organizational process monitoring in service choreographies

Validation Set-up & Result	
Name	Cross-organizational process monitoring in service choreographies
Synopsis	The validation shows that cross-organizational process monitoring can be realized based on choreography and WS technologies.
Authors	Wetzstein, B., Karastoyanova, D., Kopp, O., Leymann, F., Zwink, D.
Research questions	Cross-Partner Process Monitoring based on Service Choreographies
Scenario	Purchase Order Processing
Method	Prototype
Description	Monitoring of processes which are distributed across organizational boundaries (e.g., due to outsourcing) has to take into account that information on private processes (as modeled in executable service orchestrations) is not available due to privacy issues. We present an approach which enables creating cross-organizational monitoring solutions in service choreographies. Thereby, partners agree on monitoring properties they provide and request in a monitoring agreement. A monitoring agreement specifies process events each partner has to expose, how they can be exchanged, correlated and aggregated in order to calculate process metrics which can serve as basis for the definition of service level agreements.
Goal	The goal of this validation is to show that the proposed cross-organizational process monitoring solution and the purchase order processing scenario (including monitoring scenarios across partners) can be realized based on WS-BPEL, BPEL4Chor and WS technologies.
Set-up	A prototype has been implemented based on the ODE BPEL process engine. It has been extended to support monitoring of service choreographies based on BPEL4Chor, modeling of monitoring agreements, and exchange of events using WS-Notification.
Inputs	As input serve a prototype including ODE BPEL process engine, ESPER CEP Framework, and WS-* technologies. The Purchase order scenario has been implemented: each choreography participant's abstract process has been refined to an executable WS-BPEL orchestration and deployed on a separate engine.
Outputs	The purchase order scenario has been successfully deployed and run on the prototype. Several process engines have separately executed their processes and exchanged the events as specified in the monitoring agreement. The specified composite events in the monitoring agreement have been evaluated correctly.
Outcome	Positive
Experiences	The performed validation confirmed the applicability of the introduced approach and was therefore very reasonable.
References	Wetzstein, B.; Karastoyanova, D.; Kopp, O.; Leymann, F.; Zwink, D.: Cross-Organizational Process Monitoring based on Service Choreographies. In: Proceedings of the 25th Annual ACM Symposium on Applied Computing (SAC 2010); Sierre, Switzerland, 21-26 March (2010)
Glossary	Service Choreography, Cross-Organizational Process Monitoring, Monitoring Agreement
Keywords	WS-BPEL, BPEL4Chor, Cross-Organizational Monitoring

3.1.8 Validation of an approach to prevention of SLA Violations via Fragment Substitution

In order to prevent SLA violations in process instances while they are still running, we have presented an integrated prediction and adaptation approach based on runtime fragment substitution. Table 12 presents a validation which shows based on a prototype implementation and a purchase order processing scenario that dynamic weaving of fragments during process runtime introduces only a small performance overhead and thus has a negligible impact on the duration of most processes.

Table 12 Prevention of SLA Violations via Fragment Substitution

Validation Set-up & Result	
Name	Prevention of SLA Violations via Fragment Substitution
Synopsis	The performance analysis shows that dynamic weaving of fragments does not introduce a big overhead ([45 : 80] ms) and thus has a negligible impact on the duration of most processes.
Authors	Leitner, P., Wetzstein, B., Karastoyanova, D., Hummer, W., Dustdar, S., Leymann, F.
Research questions	Runtime Prediction of KPIs and SLA Violations Based on Machine Learning Techniques; Adaptation of QoS-aware Service Compositions based on Influential Factor Analysis and Prediction
Scenario	Purchase Order Processing
Method	Prototype, Experiments
Description	In this work we show how aspect-oriented programming can be used for runtime adaptation of service compositions in order to prevent SLA violations. Adaptations are triggered by predicted violations (based on machine learning techniques), and are implemented as fragment substitutions in the service composition. Fragments are full-edged standalone compositions which implement some part of business logic, and are linked into the original composition via virtual activities. Before substitution alternative fragments are evaluated with respect to their expected impact on the performance of the composition, and those fragments which are best suited to prevent a predicted violation are chosen.
Goal	The adaptation (weaving) is performed while the instance is running and thus has an impact on the duration of the instance (in case of offline weaving). The goal of this validation is to show that the proposed adaptation approach using offline weaving has a small impact on the duration of the process.
Set-up	The prototype has been implemented based on the Windows Workflow Foundation technology.
Inputs	We monitor the average execution time of dynamically weaved composition instances with an increasing number of activities, and compare them to the same instance defined statically. We also compare online and offline weaving. To minimize external influences all results are the average of 50 independent test runs.
Outputs	The outcomes of the experiments are as follows: Online weaved compositions have a very little overhead as compared to statically defined compositions. Offline weaving introduces overhead, as the composition has to be suspended, while the fragments are selected and weaved into the composition. The largest part of these factors is the actual weaving time. Interestingly, offline weaving is faster than online weaving (since, in the online case, some additional sanity checks are done by the Windows Workflow runtime). However, this increased weaving time for online weaving does not matter too much, since the online weaving time does not directly impact the execution time of the process instance. Overall, the overhead introduced by weaving is relatively constant in [45:80] ms, even for large fragments with more than 80 activities.
Outcome	Positive
Experiences	The performed validation confirmed the applicability of the introduced approach and was therefore very reasonable.
References	Leitner, P.; Wetzstein, B.; Karastoyanova, D.; Hummer, W.; Dustdar, S.; Leymann, F.: Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution. In: Proceedings of the 8th International Conference on Service Oriented Computing (ICSOC 2010). Springer (2010)

Glossary	Service Composition, Process Fragment, SLA Prediction, Prevention
Keywords	SLA Prediction, AOP, Process Fragment

3.1.9 Validation of Adaptation Needs through Proactive Identification

The proactive identification of adaptation needs helps to automatically and dynamically adapt to changing conditions and changes of service functionality and quality in SBA. This is of interest especially where applications are composed from third party services, which are not under the control of the service consumer. Table 13 reports the validation of this approach based on a prototypical implementation.

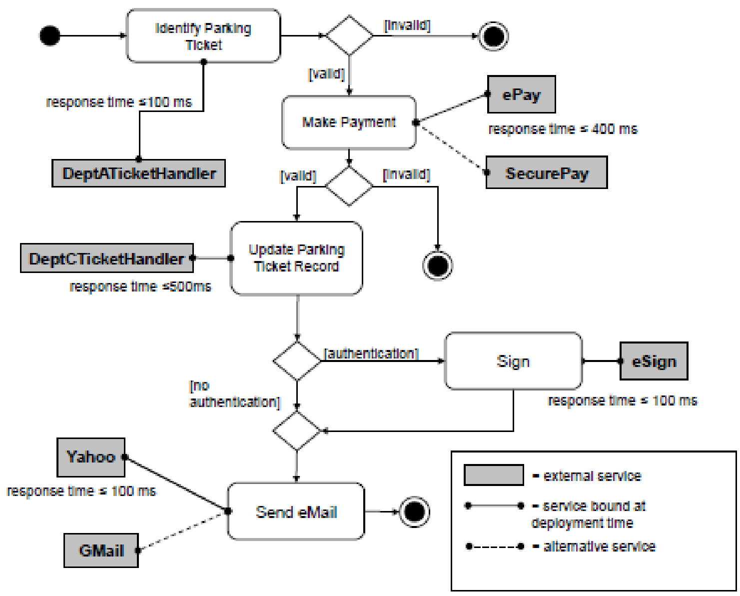
Table 13 Proactive Identification of Adaptation Needs

<table>
<tr><th colspan="2">Validation Set-up & Result</th></tr>
<tr><td>Name</td><td>Proactive Identification of Adaptation Needs</td></tr>
<tr><td>Synopsis</td><td>The approach allows the proactive identification of adaptation needs, when executing workflows (e.g. BPEL-workflows). This approach is validated by an implementation.</td></tr>
<tr><td>Authors</td><td>Metzger, A., Schmieders, E., Cappiello, C., Di Nitto, E., Kazhamiakin, R., Pernici, B., Pistore, M.</td></tr>
<tr><td>Research questions</td><td>

- 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

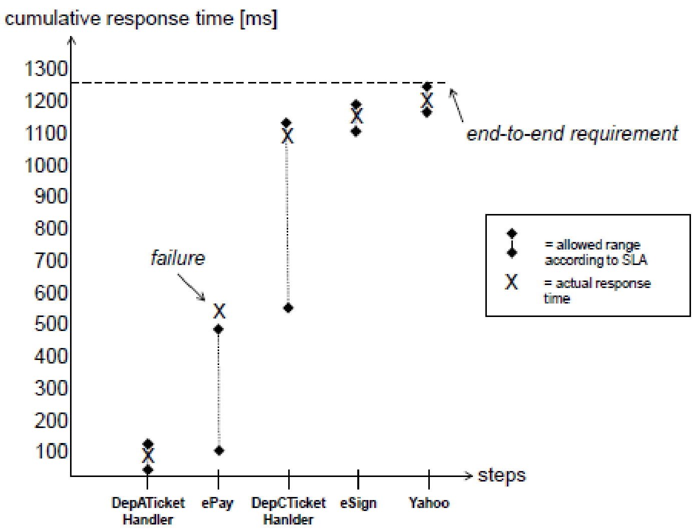
</td></tr>
<tr><td>Scenario</td><td>

We elaborate a “parking-ticket”-BPEL-workflow based on the eGovernment domain. The workflow as well as the service composition of the eGovernment application are depicted in the figure below as an extended activity diagram.



The gray boxes denote concrete services that can be composed to an eGovernment application. In the example, each service is provided by a third party. Solid connections between workflow actions and services denote the bindings established at deployment time. Dashed connections denote possible alternative services (from a different provider). In addition, the diagram is annotated with information about the negotiated response times (which could be stipulated by means of SLAs). We assume that the overall workflow is expected

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

	to have a response time of at most 1250 ms. This quality requirement can be satisfied by the bound services, provided that they meet their negotiated maximum response.
Method	Prototype
Description	Adaptive SBAs automatically and dynamically adapt to changing conditions and changes of service functionality and quality. Common monitoring approaches are problematic, as they trigger adaptations as soon as deviations occur. This might lead to unnecessary adaptations (cf. Figure below). In order to avoid unnecessary and costly adaptations, the adaptation needs have to be identified more precisely. To enable an automatic adaptation, the relevant artifacts, as well as the properties of the SBAs and their context need to be formalized to make them amenable to automated checks and decisions.  Based on these artifacts the approach identifies the need for an adaptation proactively.
Goal	The goal was to show the producibility of the approach, implementing a prototype. Furthermore, the validation should support the process model suggested by the approach, defining when and how the formalization and the techniques should be applied when developing, evolving and adapting service-based applications.
Set-up	For evaluating the process model proposed by the approach, we specify the artifacts along the S-Cube Service Life-Cycle. We specify the QoS requirement “Application must terminate after 1250 ms” by using ALBERT. We develop the BPEL workflow, based on the scenario introduced above. Finally, we again use ALBERT, this time for specifying the assumptions. We use the BOGOR model checker, which identifies the need for adaptation. We run the validation on a Windows 7/x86 platform.
Inputs	We use a BPEL process, which we specify in BIR, the input language for the BOGOR model checker. We use BOGOR to check the BPEL process against its QoS requirement.
Outputs	First, applying the proposed process successfully to the parking ticket scenario (introduced above) supports the hypothesis that the process is applicable to construct a system which can adapt proactively. Second, successfully instrumenting the BOGOR model checker proves, that the need for adaptation can be identified early in time.
Outcome	positive
Experiences	We are confident that those techniques will become especially relevant in the setting of the “Internet of Services”, where applications will increasingly be

	composed from third party services, which are not under the control of the service consumer. This implies that applications and their constituent services need to be continuously checked during their operation such that they can be dynamically adapted or evolved in order to respond to failures or unexpected changes of third party services.
References	Metzger, A., Schmieders, E., Cappiello, C., Nitto, E.D., Kazhamiakin, R., Pernici, B., Pistore, M.: Towards proactive adaptation: A journey along the s-cube service life-cycle. In: Maintenance and Evolution of Service-Oriented Systems (2010)
Glossary	Proactive Adaptation, Adaptation Trigger , Assumption, Service Orchestration, Monitoring, Adaptation, Workflow, Service Runtime
Keywords	Proactive Adaptation, Adaptation Trigger , Assumption

3.1.10 Validation of Web Service selection

The validation of web service selection based on precision and recall is reported in Table 14. The validations shows, that the approach leads to a proper fit of functional and non-functional users' requirements with high accuracy and efficiency. Therefore, the validation indicates that the approach can be used to select automatically relevant and high QoS services.

Table 14 Empirical validation of Web service selection in terms of precision and recall

Validation Set-up & Result	
Name	Empirical validation of Web service selection in terms of precision and recall.
Synopsis	The validation shows that our approach allows the selection of Web services that fit closely functional and non-functional users' requirements with high accuracy and efficiency.
Authors	Driss, M., Moha, N., Jamoussi, Y., Jézéquel, J.-M., Ghézala, H. H. B.
Research questions	Modeling of service-based applications in terms of functional and non-functional users' requirements. Discovery of relevant Web services that fit closely functional users' requirements. Selection of relevant and high QoS Web services.
Scenario	E-books service-based application: this application allows books search and acquisition, shipment organization, and payment finalization.
Method	Empirical validation based on two metrics: precision and recall (Prototype)
Description	The experimental results on the e-books service based application shows that our approach allows the selection of relevant and high QoS services with high accuracy (the average precision is 89.41%) and efficiency (the average recall is 95.43%).
Goal	The goal of the validation is to show that FCA can be applied to select automatically relevant and high QoS services. FCA provides clear and organized structures called concept lattices of potential services to enable users to easily check out services that satisfy their functional and non-functional requirements.
Set-up	Empirical proof in terms of precision and recall: Precision= $\frac{\lvert\{\text{true relevant and high QoS operational services}\}\cap\{\text{returned operational services}\}\rvert}{\lvert\{\text{returned operational services}\}\rvert}$ Recall =

$$\frac{|\{\text{true relevant and high QoS operational services}\} \cap \{\text{returned operational services}\}|}{|\{\text{true relevant and high QoS operational services}\}|}$$

Inputs	Atomic intentional services of the e-books service-based application (29 services).
Outputs	Experimental results related to 10 intentional services of the e-books service-based application.

Intentional services	Service-Finder			Our Approach			Precision	Recall
	Keywords	Returned operational services	True relevant and high QoS operational services	Discovery	Selection	True relevant and high QoS operational services		
$ab1_i \rightarrow S_{Search\ a\ book\ by\ ISBN}$	'search + book + isbn'	7	2/7	4/7	2/4	2/2	2/2 (100%)	2/2 (100%)
$ab2_i \rightarrow S_{Search\ a\ book\ by\ author(s)}$	'search + book + author'	16	3/16	10/16	4/10	3/4	3/4 (75%)	3/3 (100%)
$ab3_i \rightarrow S_{Search\ a\ book\ by\ title}$	'search + book + title'	31	5/31	22/31	6/22	5/6	5/6 (83.33%)	5/5 (100%)
...	...	...	...	...	...	...	...	...
$cc2_i \rightarrow S_{Sort\ books\ by\ price}$	'sort + price'	64	11/64	52/64	9/52	8/9	8/9 (88.89%)	8/11 (72.73%)
$cc3_i \rightarrow S_{Sort\ books\ by\ seller\ location}$	'sort + location'	58	12/58	40/58	11/40	11/11	11/11 (100%)	11/12 (91.67%)
...	...	...	...	...	...	...	...	...
$ee1_i \rightarrow S_{Change\ the\ currency}$	'change + currency'	55	8/55	30/55	7/30	6/7	6/7 (85.71%)	6/8 (75%)
$ab1_{i+1} \rightarrow S_{Create\ an\ account}$	'create + account'	330	26/330	203/330	25/203	22/25	22/25 (88%)	22/26 (84.62%)
$ab2_{i+1} \rightarrow S_{Load\ an\ account}$	'load + account'	29	4/29	20/29	5/20	4/5	4/5 (80%)	4/4 (100%)
$bc1_{i+1} \rightarrow S_{Pay\ with\ credit\ card}$	'payment + credit + card'	77	4/77	16/77	4/16	4/4	4/4 (100%)	4/4 (100%)
$cd1_{i+1} \rightarrow S_{Send\ sms}$	'send + sms'	162	17/162	105/162	18/105	15/18	15/18 (83.33%)	15/17 (88.24%)
						Average	89.41%	95.43%

In this table, the first column corresponds to intentional services. In the second column, we list, first, the keywords used to query Service-Finder; then, the number of services returned by Service-Finder; and finally, the number of services identified manually as true relevant and high QoS services. In the third column, we enumerate the number of services obtained, first, after the discovery step of our approach, and then, after the selection step, and finally the number of true services among those returned after selection step. The two last columns correspond to the precision and the recall. For example, the two keywords of the intentional service $S_{Send\ sms}$ are: 'send + sms'. The query returns an initial set of 162 operational services. Among this set, only 17 services are verified manually as true relevant and high QoS services. The discovery step reduces the initial set to a second set of 105 services. The selection step using FCA reduces the second set to 18 services. Among these services, only 15 services are verified manually as true relevant and high QoS services. The precision of $S_{Send\ sms}$ is 83.33% and the recall is 88.24%.
The table shows that the precision and recall of our approach are both very high. The average precision is 89.41% and the average recall is 95.43%.

Outcome	Positive.
Experiences	Experimental results show that our approach allows the selection of relevant and high QoS operational services with high accuracy and efficiency. To increase the robustness of our approach, we need to use more advanced semantic techniques for the filtration of the discovered operational services. Moreover, our selection is based only on two QoS properties that are availability and response time. Thus, we need to

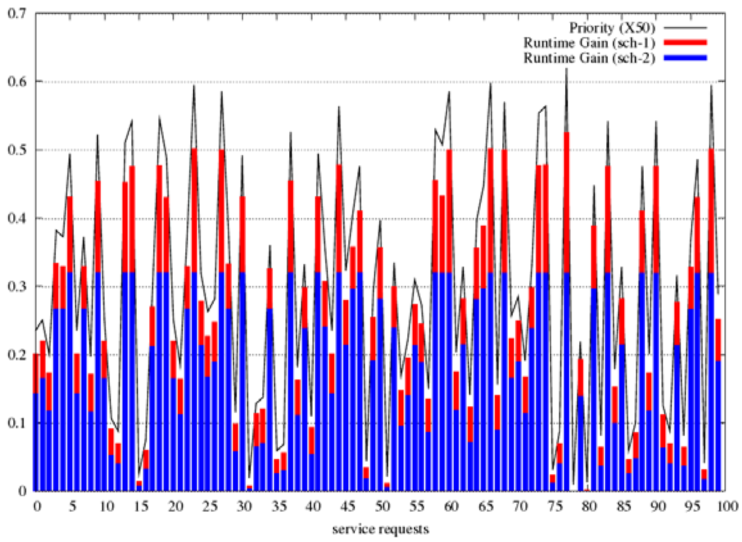
	enhance our selection method with multiple QoS properties to identify more efficiently high QoS services.
References	Driss, M., Moha, N., Jamoussi, Y., Jézéquel, J.-M., Ghézala, H. H. B.: A Requirement-Centric Approach to Web Service Modeling, Discovery, and Selection. In: Proceedings of the 8th International Conference on Service Oriented Computing ICSOC 2010, pp. 258-272. Springer (2009)
Glossary	Service-based Applications (SBAs), Quality of Service (QoS), Formal Concept Analysis (FCA).
Keywords	Service-Based Applications, Users' Requirements modeling, Service Discovery, Service Selection, QoS, Formal Concept Analysis.

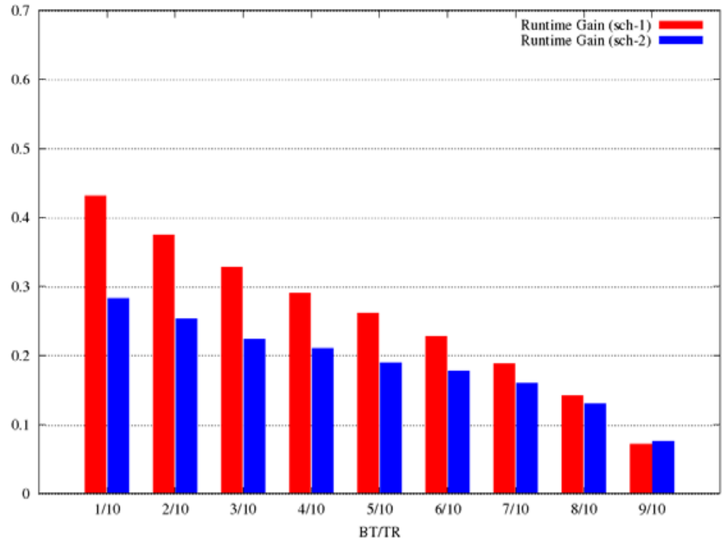
3.1.11 Validation of adaptable service scheduling

For a flexible service scheduling, this approach proposes the use of continuations to control service execution in order to adapt scheduling resources allocated for the service execution according to user requests. Table 15 reports on the validation of this approach in terms of the viability and flexibility of a continuation-based mechanism to control service execution.

Table 15 Validation of adaptable service scheduling

Validation Set-up & Result	
Name	Validation of adaptable service scheduling.
Synopsis	The continuation-based mechanism for service scheduling at application-level is validated highlighting how the mechanism can be used to implement an economy-based service scheduling that adapts resource allocation for service execution according to user requests and system workload.
Authors	Di Napoli, C., Giordano, M.
Research questions	Self-optimization and self-healing of a single service. On-demand, dynamic service provisioning.
Scenario	Economy-based service provisioning (no reference to a specific S-Cube scenario).
Method	Prototyping and experiments with the prototype.
Description	We propose the use of continuations to control service execution in order to adapt scheduling resources allocated for the service execution according to user requests. We argue that the proposed approach provides a flexible programming mechanism to implement a service dynamic scheduling at application level. The behaviour of two cost-based resource sharing scheduling policies is analyzed.
Goal	The goal of the validation is to show the viability and flexibility of a continuation-based mechanism to control service execution for the specification/implementation of scheduling policies at application level.
Set-up	The prototypical evaluation was carried out with a prototype of a continuation-based service provider (developed for the purpose) equipped with a scheduler module that implements different scheduling policies. Simulations were carried out to analyze the behaviour of two cost-based resource sharing scheduling policies by varying scheduling parameters (service pre-emption frequencies). The simulations measure the service response time with a set of incoming service execution requests with a uniform distribution of the service costs (that the user is willing to pay for the service execution when the request is submitted). Experiments show 1) how close the distribution of

	priorities assigned by the scheduler approximates the service cost distribution; 2) how fast each policy adapts service priority when cost-update requests are received during service execution.
Inputs	We define the improvement of the service execution time, as the Runtime Gain (RG) given by: $RG_i = 1 – TR/ET_i$, where TR is the Total Runtime, i.e. the elapsed time from the submission of the set of requests to the end of the execution of the last service, and ET_i is the Elapsed Time from the submission of the set of requests to the end of the execution of the service i. For experiment 1) we measure the RG_i distribution for the two scheduling policies varying the number of incoming requests. For experiment 2) we measure the variation of the RG_i for one service in the set for which a cost-update (to the max value allowed) is issued varying the time when the update occurs during service execution (Boost Time). All the other services in the set have the same cost.
Outputs	For experiment 1) the RG_i distribution for the two scheduling policies is plotted together with the distribution of the priorities assigned by the scheduler. The results show that for one policy the RG_i distribution is closer to the priority one privileging the execution of higher cost services (see Figure 1). For experiment 2) the RG_i of the service for which the cost-update occurred increases by decreasing the boost time (BT) as expected. One policy reacts faster to the update request than the other because it eagerly privileges higher cost services (see Figure 2).  Figure 1: RG_i distribution for the two policies

	 Figure 2: RG for a service requiring a cost update
Outcome	Positive.
Experiences	The experimental results are encouraging and they need to be tested in more practical scenarios.
References	Di Napoli, C., Giordano, M.: Experimenting Scheduling Policies with Continuation-based Services. In: Proceedings of Int. Symposium on Advanced Topics on Information Technologies and Applications ITA 2010 (in conjunction with CIT2010), pp. 1498-1503. IEEE Computer Society Press (2010)
Glossary	Adaptation, Service Runtime Management Process.
Keywords	Continuations, Economy-based service scheduling

3.2 Validation Method: Experiment

3.2.1 Validation of prediction-based adaptation

Virtual Engineering requires the execution of workflows where single jobs are executed in the Cloud. In order to appropriate schedule this execution, predictions based on historical data are used to predict QoS metrics and to proper allocate the resources needed for the execution. Table 16 documents the positive validation of a laboratory experiment utilizing a prototype that uses predictions to successfully adapt job schedules, gives details on job executions and indicates error rates in the predictions.

Table 16 Adaptation based on predictions using historical data

Validation Set-up & Result	
Name	Adaptation based on predictions using historical data
Synopsis	The use of predictions in adaptation mechanisms is validated in a Cloud Computing scenario, using a framework exploiting semantically enhanced historical data for predicting the behaviour of tasks and resources in the system, and allocating the resources according to these predictions
Authors	Ejarque J., Micsik, A., Sirvent, R., Pallinger, P., Kovacs, L. and Badia, R. M.
Research questions	Runtime Prediction of KPIs and SLA Violations Based on Machine Learning Techniques
Scenario	Virtual Engineering requires the execution of high-performance workflows. In the scenario, each step (job) of the workflow is performed in a Cloud environment, where the jobs have to be scheduled to be executed on hosts.

Method	Experiment, Prototype
Description	We show that based on collected historical data it is possible to generate useful predictions on QoS metrics of service executions. These predictions are used in the adaptation of existing job schedules. Furthermore, historical data is processed and stored using a semantic format (RDF).
Goal	To present the feasibility of the approach.
Set-up	Settings: a test grid environment in a lab Tools used: test environment built using Java, Jena, Jade, Weka, etc.
Inputs	Sample Virtual Engineering workflows (airflow simulations), semantic descriptions of computational resources, historical job execution data
Outputs	Workflow execution results, details of job executions, successfully adapted job schedules, error rates in predictions
Outcome	positive
Experiences	Available in the paper.
References	Ejarque, J., Micsik, A., Sirvent, R., Pallinger, P., Kovacs, L., Badia, R. M.: Semantic Resource Allocation with Historical Data Based Predictions. In: Proceedings of the First International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2010. Lisbon, Portugal. IARIA (2010)
Glossary	Quality of Service-Based Adaptation, Self-Adaptation, Grid Scheduling
Keywords	multi-agent, semantics, scheduling, resource allocation, historical data, predictions, grid computing, cloud computing, distributed systems

3.2.2 Validation of context prediction for service availability in dynamic environments

The validation reported in Table 17 shows the accuracy and efficiency of a context prediction system applied for the prediction of service availabilities. The experiment observes the specification of an application-specific prediction model as a basis for runtime prediction of values which are relevant for a context-aware execution of service compositions.

Table 17 Validation of context prediction for service availability in dynamic environments

Validation Set-up & Result	
Name	Validation of context prediction for service availability in dynamic environments
Synopsis	The validation shows accuracy and efficiency of a context prediction system applied for the prediction of service availabilities. The result is derived by a series of experiments based on the current prototype implementation of the prediction framework.
Authors	Hamann, K., Zaplata, S., Lamersdorf, W.
Research questions	Context-Aware Execution of Distributed Processes.
Scenario	-
Method	Experiments
Description	The novel approach of Structured Context Prediction (SCP) allows for the specification of an application-specific prediction model as a basis for runtime prediction of values which are relevant for a context-aware execution of service compositions. Using the example of service availability, an exemplary prediction model has been developed and has been tested in terms of performance and accuracy. Furthermore, a long-time study was executed in order to test the overall scalability of the approach. The validation shows that high accuracy and efficiency are possible by using a hybrid application of prediction mechanisms which is based on

	the expected domain-specific structure of variable dependencies. The results emphasize that such context prediction approach can be applied to support runtime decisions for process execution in short time and with linear memory requirements. Thus, predictions based on this approach are even applicable for dynamic environments including e.g. resource-restricted mobile devices.
Goal	The goal of the validation is to show that it is possible to specify an adequate prediction model based on a generic prediction framework which1) shows acceptable results regarding accuracy and efficiency of context predictions, and 2) allows for runtime context predictions.
Set-up	The validation is based on an executable prototype of the prediction framework for Structured Context Prediction (SCP). The prototype and the evaluation were run on an average notebook (1.5 GHz, Pentium M processor).
Inputs	An application-specific prediction model holds all necessary variable dependencies in order to predict the availability of a service at a specific time and at a specific location. The model uses a hybrid application of prediction mechanisms consisting of probability tables, linear regression and majority vote. Based on this prediction model, the tested example consists of realistic historical data about the behaviour of a user and its mobile device spanning an interval of seven days (Monday to Sunday). It contains the net size, the time of day, the position and the service availability as values of the corresponding variables at different points of time, representing context data measured by real sensors. For the first practical experiments, two services with different behaviours have been chosen: A stationary printer service is regularly available when the user is at work (Monday to Friday, at daytime). An ad-hoc file exchange service is offered spontaneously by few mobile devices carried by other people in the direct vicinity of the user and is thus only available very infrequently.
Outputs	Efficiency: Using the methods of the configuration as specified above, the memory requirements are bounded and do not significantly increase because the instance data, i.e. the knowledge learned by the prediction methods, is saved instead of measured raw context data. Here, the memory consumption is dominated by probability tables. The maximum amount of memory required for the instance data in the example is about 20 KB and the processing time for learning is insignificant (i.e. considerably less than 1% CPU load). Theoretical considerations show the time complexity is linear. This result is also confirmed by the practical experiments regarding the number of time steps in the time interval and similarly the number of prediction rounds. The results indicate that also such relatively complex predictions take less than one second and, thus, the resource consumption is relatively well suited even for average or less powerful mobile devices (e.g. smartphones). Accuracy: Because the (more simple) ad-hoc file exchange service is often unavailable, this regularity can be learned quickly and predictions about its availability already start with relatively good results, i.e. predicting that the service is not available is correct in most cases. Furthermore, in the following days, the system learns to distinguish the availability of the service and the accuracy slightly increases. Scalability: The developed prediction net provides the possibility that parts of the learned knowledge can be re-used, so that only little

	information has to be saved for each new service detected at runtime. The long term application of the configured prediction system shows that the service-independent knowledge is not influenced by the number of detected services and consumes about 100 kilobyte for an advanced version of the prediction net.
Outcome	Positive.
Experiences	The experiments show that the domain-specific prediction model achieves as well high accuracy as efficiency. However, it cannot be determined if all possible domains can be supported by an appropriate prediction model this way. Thus, advanced methods and appropriate tools to support the evaluation of each prediction models' correctness and effectiveness have to be developed.
References	Meiners, M., Zaplata, S., Lamersdorf, W.: Structured context prediction: A generic approach. In: Eliassen, F., Kapitza, R. (eds), Distributed Applications and Interoperable Systems (6115), pages 84–97. Heidelberg (2010) Zaplata, S., Meiners, M., Lamersdorf, W.: Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction. To appear in: Software – Practice and Experience (SPE), Wiley InterScience (2011)
Glossary	Service Orchestration, Process Migration, Context, Context-Awareness
Keywords	Context Prediction, Service-Availability

3.2.3 Validation of a meta-brokering approach

Table 18 first reports the validation of a grid meta-brokering approach by simulated dynamic adaptation of services on Grid. The validation is based on a general Grid simulation environment that is based on GridSim, in which all the related grid resource management entities can be simulated and coordinated. Further, Table 19 documents the validation of meta-brokering with the Pliant system. The validation basically shows that the performance of meta-brokering can be enhanced using a scheduling solution based on the Pliant system.

Table 18 Validation of Grid meta-brokering by simulation

Validation Set-up & Result	
Name	Validation of Grid meta-brokering by simulation
Synopsis	The validation shows that the interoperable meta-brokering solution of GMBS achieved an order of magnitude better performance in Grid application execution compared to the general, non-interoperable Grid utilization.
Authors	Kertesz, A., Kacsuk, P.
Research questions	Supporting adaptation of service-based applications
Scenario	
Method	experiment
Description	In order to evaluate the proposed meta-brokering service, we have created a general Grid simulation environment based on GridSim, in which all the related grid resource management entities can be simulated and coordinated, e.g. evaluating VO-based resource allocation and dynamic resource provisioning techniques in global Grids.
Goal	The goal of the validation is to show that the makespan of application executions in multi-Grid environments can be reduced by the concept of meta-brokering.
Set-up	Different simulated Grid environments have been set up ranging from 6 to 14 brokers using 24 to 48 resources fed by real workload logs during the

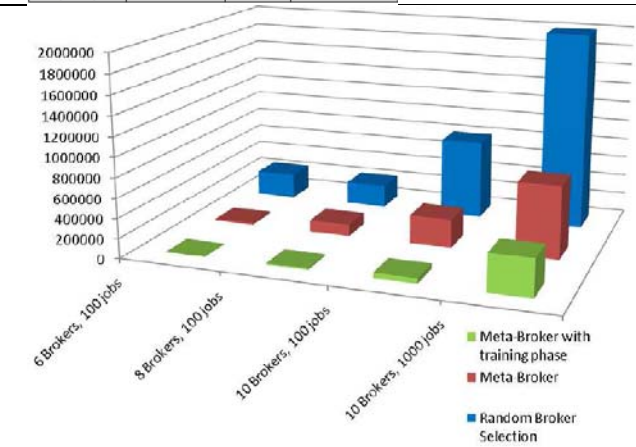
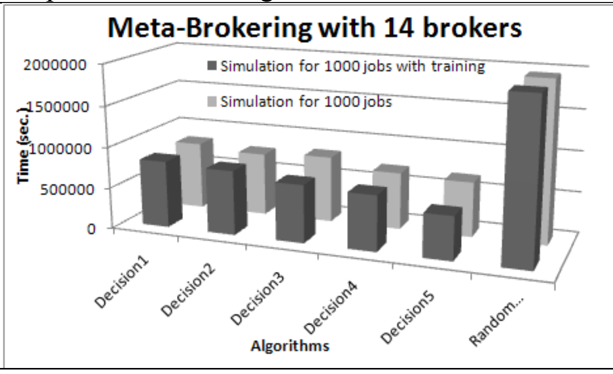
<table>
<tr><td></td><td colspan="2">evaluations.</td></tr>
<tr><td>Inputs</td><td>
<table>
<tr><th>Brokers</th><th>Resour-ces</th><th>Jobs</th><th>Work-load</th></tr>
<tr><td>6/X (fcpu)</td><td>6x8(x2)</td><td rowspan="3">100</td><td>20x(6x8)</td></tr>
<tr><td>3/X (nfail)</td><td>3x10(x2)</td><td>20x(3x10)</td></tr>
<tr><td>1/X (rnd)</td><td>1x16(x2)</td><td>20x(1x16)</td></tr>
<tr><td>6/X (fcpu)</td><td>6x8(x2)</td><td rowspan="3">1000</td><td>50x(6x8)</td></tr>
<tr><td>3/X (nfail)</td><td>3x10(x2)</td><td>50x(3x10)</td></tr>
<tr><td>1/X (rnd)</td><td>1x16(x2)</td><td>50x(1x16)</td></tr>
</table>
</td><td>This table is an example for two environment set-ups with 100 and 1000 input jobs per simulation. The input workload jobs are submitted as background load to the utilized resources (20 and 50 per resource respectively).</td></tr>
<tr><td>Outputs</td><td>
</td><td>This figure summarizes the measured makespan for the different experiments.</td></tr>
<tr><td>Outcome</td><td colspan="2">positive</td></tr>
<tr><td>Experiences</td><td colspan="2">Available in the paper</td></tr>
<tr><td>References</td><td colspan="2">Kertesz, A., Kacsuk, P.: GMBS: A new middleware service for making grids interoperable. In: Journal of Future Generation Computer Systems, 26(4), pp. 542 – 553. Elsevier (2010)</td></tr>
<tr><td>Glossary</td><td colspan="2">Grid brokering</td></tr>
<tr><td>Keywords</td><td colspan="2">Grid interoperability, meta-brokering</td></tr>
</table>

Table 19 Validation of meta-brokering with the Pliant system

Validation Set-up & Result	
Name	Validation of meta-brokering with the Pliant system
Synopsis	The validation shows that the performance of meta-brokering can be enhanced using a scheduling solution based on the Pliant system
Authors	Dombi, J. D., Kertesz, A.
Research questions	Supporting adaptation of service-based applications
Scenario	
Method	experiment
Description	In order to evaluate the proposed meta-brokering service, we have created a general Grid simulation environment based on GridSim, in which all the related grid resource management entities can be simulated and coordinated, e.g. evaluating VO-based resource allocation and dynamic resource provisioning techniques in global Grids.
Goal	The goal of the validation is to show that the makespan of application executions in multi-Grid environments can be further reduced by scheduling using the Pliant system.
Set-up	The simulated Grid environment had 14 brokers using 4 to 12 resources fed by real workload logs during the evaluations.

Validation Set-up & Result	
Inputs	1000 input jobs have been submitted in each experiment using different scheduling parameters. 50 workload jobs have been submitted as background load per resource during each simulation.
Outputs	 This figure summarizes the measured makespan for the experiments using different scheduling parameters.
Outcome	positive
Experiences	Available in the paper
References	Dombi, J. D., Kertesz, A.: Scheduling solution for grid meta-brokering using the pliant system. In: Proceedings of the 2nd International Conference on Agents and Artificial Intelligence. Springer (2010) http://www.inf.u-szeged.hu/~keratt/papers/mb_sched_ICAART_2010.pdf
Glossary	Grid brokering, Grid scheduling
Keywords	Pliant system, Sigmoid function, Meta-brokering.

3.3 Validation Method: Case Study

3.3.1 Validation of a two-tier architecture and ad-hoc management

Table 20 describes the validation of a two-tier architecture for monitoring and ad-hoc management of distributed business processes. The validated architecture is compared with two customary approaches. The validation has shown that these customary approaches are not sufficient to properly deal with ad-hoc management of distributed business processes.

Table 20 Validation of a two-tier architecture for monitoring and ad-hoc management of distributed business processes

Validation Set-up & Result	
Name	Validation of a two-tier architecture for monitoring and ad-hoc management of distributed business processes
Synopsis	This validation evaluates the two-tier architecture for monitoring and ad-hoc management of distributed business processes. Using an example scenario, it compares this novel approach with two customary approaches, i.e., weaving of monitoring activities, and event-based monitoring.
Authors	Hamann, K., Zaplata, S., Lamersdorf, W.
Research questions	Context-Aware Execution of Distributed Processes Monitoring of Business Transactions Management of Distributed Service Compositions
Scenario	The validation uses an example scenario from the *eErasmus eHigher Education* (eEH) project. In order to facilitate a uniform exchange of students joining this program, a standardized process is proposed for participating universities. The simplified functional process used here involves subcontracting the host university for *approving the credentials* necessary for taking courses there, allowing to take *courses and exams* until a specified deadline and *preparing the*

	credentials achieved at the host university in order to acknowledge them at the home university. The distributed execution involves several management requirements which are *expected in advance*, i.e. before execution of the process starts: (R1) The host university is paid a certain amount of money for each student and for the associated administration effort. Therefore, the duration of each activity executed by the host university has to be logged. (R2) In order to handle potential errors in time, the home university wants to be sure that the foreign university has received the sub-process and is able to execute it, and, (R3) if duration of an activity expected as critical (here *preparation of credentials*) exceeds the average time for executing a task, (R4) the activity should be skipped in order to at least allow the control flow of the process to return to the calling system. (R5) As it sometimes happens that the deadline for taking courses is adapted by the host university, e.g. because the student gets ill, the home university wants to know about such events in order to avoid automatic removal from the home register of students. In addition, there are a number of *unexpected occurrences* during the runtime of this rather long-running (i.e. several months) process: First, a financial aid program asks about the status of the student's overall study (R6). Second, the student has married and his/her name has to be adapted (R7).
Method	Scenario (Case Study)
Description	The validation has shown that the two customary approaches to monitor the execution of business processes cannot deal with ad-hoc management requirements on distributed processes. However, the proposed architecture provides means to gain information about the status of a process execution and also to control the execution, both in an ad-hoc manner.
Goal	The goal of this validation is to show that the two-tier architecture for monitoring and ad-hoc management of distributed business processes is applicable is real scenarios and that, in contrast to usual approaches, the proposed architecture can handle also unexpected occurrences of monitoring and management requirements.
Set-up	The validation evaluates the ad-hoc management architecture conceptually using the example scenario. The proposed architecture is compared with two customary approaches, i.e., weaving of monitoring activities, and event-based monitoring.
Inputs	Example Scenario with seven requirements for monitoring and management capabilities.
Outputs	It shows that monitoring aspects which are known in advance can be realized by the design time insertion of respective monitoring activities (timer activities and passing of variables values to the central monitoring service) and by the event-based monitoring and the ad-hoc management approach (by subscription of the respective events). The detection of abnormal activity duration can be realized by the ad-hoc management as a complex rule involving also additional information about previous process instances executed on this system and calculating their average time of execution. This is neither possible by a system which makes use of events only (the events of other process instances have not been captured before) nor by activity weaving (histories of other process instances are not visible in the monitored process instance). Skipping critical activities is also a problem, because event-based monitoring does not offer control functionalities at all, and activity weaving cannot skip crashed activities by weaving an "end activity" because in this case control flow will not reach this activity. Considering unexpected occurrences, the ad-hoc management shows its biggest advantage: The status retrieval can be made by calling the process's resource property process status and interesting data values directly. Both activity weaving and event-based monitoring can provide this data only in case a

	monitoring activity is inserted after each functional activity resp. all available events have been subscribed. Therefore, it is more or less a coincidence if such requests can be fulfilled as they cannot be determined in advance and relevant properties have to be weaved/subscribed before runtime. The ad-hoc variable modification is also not possible because of missing runtime modification operations. However, even by using the ad-hoc management approach, the process manager has to be careful not to violate the integrity of the process. Considering non-functional characteristics, it shows that desired separation between business logic and management logic can be achieved by event-based and ad-hoc management approaches (as the original business process does not have to be changed), but not by activity weaving. Especially in the context of mobile process management, the approach of activity weaving furthermore proves to be very instable (i.e. if the monitoring service is not available, the process execution is delayed or even fails). For the event-based approach, no delays affected by the management are visible at all – however no reactions are possible and thus events can be emitted in parallel to an ongoing process execution without delay. Compared with both event-based and ad-hoc management approaches, activity weaving has, however, the advantage that no system modifications, security mechanisms or agreements are necessary. As summary of the results can be found in the following Table:

Management requirement	**Ad-hoc management**	**Event-based monitoring**	**Activity weaving**
R1: Duration of activities	+	+	+
R2: Instance started	+	+	+
R3: Detect critical activity duration	+	○	○
R4: Skip critical activity if necessary	+	-	-
R5: Observe variable value	+	+	+
R6: Ad-hoc status retrieval	+	○	○
R7: Ad-hoc variable value modification	+	-	-

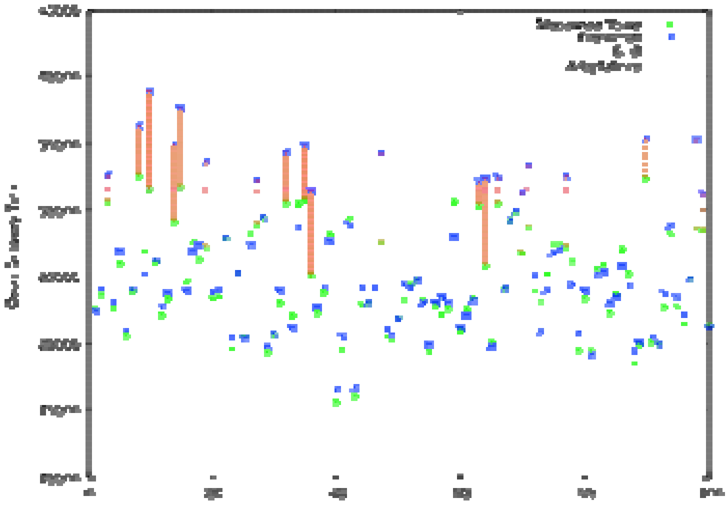
Outcome	positive
References	Zaplata, S., Straßenburg, D., Wunderlich, B., Bade, D., Hamann, K., Lamersdorf, W.: Ad-hoc Management Capabilities for Distributed Business Processes. In: 3rd International Conference on Business Process and Services Computing (BPSC 2010). Bonn (2010)
Glossary	Monitoring, Service Composition, Service Orchestration, Runtime Process Migration
Keywords	Two-Tier Architecture, Manageable Resource Management Component

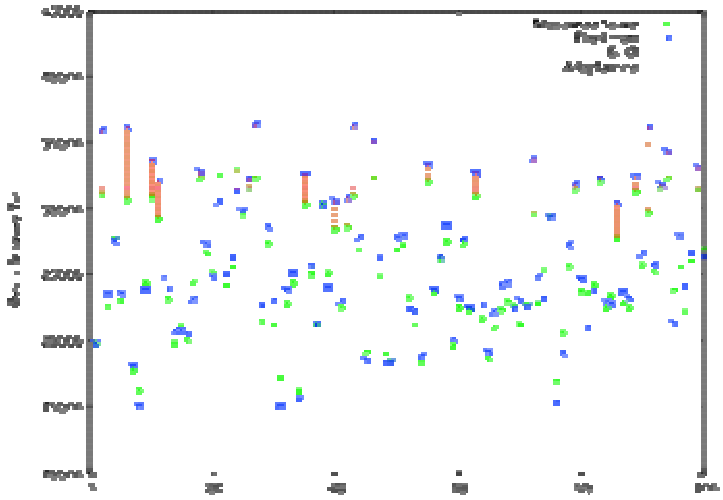
3.3.2 Validation of SLA Compliance Approach in VRESCO

The number of SLA violations can be significantly be influenced by triggering preventative adaptation actions. Table 21 shows that the SLA compliance approach used in VRESCo is able to cope with this aspect. The validation documents how many SLA violations can be prevented.

Table 21 Numerical Validation of SLA Compliance Approach in VRESCO

Validation Set-up & Result	
Name	Numerical Validation of SLA Compliance Approach in VRESCO
Synopsis	The validation shows that the SLA compliance approach used in the VRESCo prototype is able to prevent a large number of SLA violations by triggering preventative adaptation actions.

Authors	Leitner, P., Michlmayr, A., Rosenberg F., Dustdar S.
Research questions	Proactive Adaptation and Predictive Monitoring, Quality Prediction Techniques to Support Proactive Adaptation, Analysis and Prediction of Quality Characteristics of Service Compositions, QoS Aware Adaptation of Service Compositions
Scenario	-
Method	Case Study
Description	We show based on a case study from the assembling domain how (1) accurate predictions of SLA violations in VRESCo are, (2) how many SLA violations can be preventing using the 'minimal' strategy, and (3) how many violations can be prevented using the 'safe' strategy. We discuss the advantages and disadvantages of both strategies based on this case.
Goal	The goal of the validation was to show how usage of the VRESCo system can reduce the costs of SLA violations for providers of composite services in real-life.
Set-up	To evaluate the approach we have implemented the case study described using Windows Workflow Foundation technology. All experiments have been carried out on an Intel Xeon Dual CPU X5450 with 3.0 Ghz and 32 GByte RAM, running Windows 2007 Server. On this machine we have hosted all services used in the composition, the composition engine, VRESCo and a number of test clients. The services emulate realistic QoS behavior (e.g., the response times of services follow a Monte Carlo simulation with service-specific parameterization).
Inputs	The case study.
Outputs	The following two figures show how prevention of SLA violations works. The first figure shows the safe startegy, the second one the minimal strategy. In both figures, for every independent instance (x-axis) we have plotted the predicted value for a single service level objective (blue x) and the actual measure value (green +). The pink horizontal line is the violation threshold. Evidently, for each case where the prediction was over or very close to the threshold an adaptation has been triggered (instances with a red vertical bar). In most of these cases the actual outcome is below the violation threshold, i.e., the violation has been prevented. 

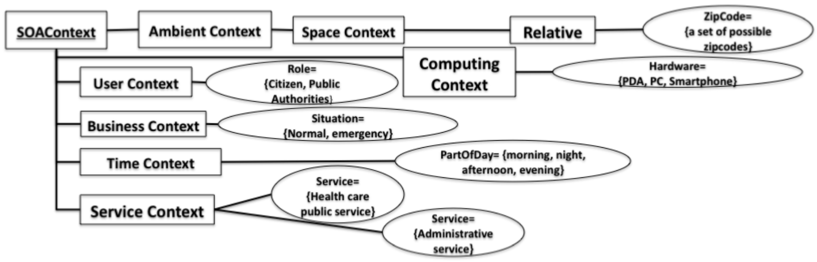
	
Outcome	Positive
Experiences	Available in the paper
References	Leitner, P., Michlmayr, A., Rosenberg, F., Dustdar, S.: Monitoring, Prediction and Prevention of SLA Violations in Composite Services. Presented at IEEE International Conference on Web Services (ICWS) Industry and Applications Track (2010)
Glossary	Self-Adaptation, Service Composition, Service Level Agreement
Keywords	Self-Adaptation, Service Composition, Service Level Agreement, Prevention of SLA Violations

3.3.3 Validation of a context-driven adaptation Process for service-based applications

The validation in Table 22 shows how a context-aware design process for service-based applications can be applied to a case study. The results witness the capability of the context-driven adaptation process to capture the key aspects of adaptation and support designers from the requirements elicitation to the construction of proper adaptation mechanisms.

Table 22 Validation of a context-driven adaptation Process for service-based applications

Validation Set-up & Result	
Name	Validation of a context-driven adaptation Process for service-based applications
Synopsis	The validation shows how a context-aware design process for service-based applications can be applied to a case study
Authors	Bucchiarone, A., Kazhamiakin, R., Cappiello, C., di Nitto, E. Mazza, V.
Research questions	Define in the life cycle phases to enable adaptation and evolution of SBA How context information could be exploited during the lifecycle Design for adaptation How can we measure, control, evaluate and improve the adaptation cycle?
Scenario	E-Government Case Study
Method	Case Study
Description	The effectiveness of the discussed principles and guidelines proposed has been evaluated by considering a real case study based on an e-government scenario. Results witness the capability of the context-driven adaptation process to capture the key aspects of adaptation and

	support designers from the requirements elicitation to the construction of proper adaptation mechanisms.
Goal	The goal of the validation is to show that the life-cycle proposed can be used as a reference tool to design and develop adaptable Service-based application.
Set-up	Definition of requirements and phases that should be considered in the design life-cycle of an adaptive SBA and definition of the context elements that are relevant for the adaptation process.
Inputs	 **The Context Model for the e-Government scenario**

Outputs

Table 1: Suitability of adaptation strategies to react to context changes

Strategy	Time	Ambient	User	Service	Computing	Business
Service substitution	X	X	X	X	X	X
Re-execution				X	X	
Re-composition				X	X	X
Fail			X	X	X	X
Service concretization	X	X	X	X	X	X
Re-negotiation			X		X	X
Compensation				X	X	
Trigger Evolution		X	X	X	X	X

Table 2: Service/Context table for the E-government scenario

Application	Time	Ambient	User	Service	Computing	Business
Health-care public services	X	X	X	X	X	X
Administrative services	X	X	X	X	X	X
Census and registration services		X	X	X	X	X
Information services	X	X	X	X	X	X
Auxiliary services		X		X	X	X

Table 3: Service/Adaptation Strategy table for the e-government scenario

Adaptation Strategy	Health-care public services	Administrative services	Census and registration services	Information services	Auxiliary services
Service substitution	X			X	X
Re-execution	X	X	X	X	X
Re-composition	X	X	X	X	X
Fail	X	X	X	X	X
Service concretization	X			X	X
Re-negotiation	X			X	X
Compensation	X	X	X		X
Trigger Evolution	X	X	X		

Outcome	Results witness the capability of the context-driven adaptation process to capture the key aspects of adaptation and support designers from the requirements elicitation to the construction of proper adaptation mechanisms. Our future roadmap includes a refinement of the adaptation process presented in this paper, its formalization, and validation. We also intend to work on the development of mechanisms and tools supporting the methodology, building on top of the actions and artifacts identified in the proposed life-cycle.
Experiences	Completely positive experience in using the context-driven adaptation process for a real case study.
References	Bucchiarone, A.; Kazhamiakin, R.; Cappiello, C.; Di Nitto, E.; Mazza, V.: A Context-driven Adaptation Process for Service-based Applications. In: Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems. ACM (2010)
Glossary	Context-Awareness, Adaptation, Service-based Applications
Keywords	Context-Awareness, Adaptation

3.3.4 Validation of Live goals for adaptive service compositions

The case study reported in Table 23 validates a methodology to link requirements with monitoring and adaptation strategies used at runtime. It proposes to enrich existing goal models with adaptive goals, responsible for the actual evolution/adaptation of the goal model at runtime. The validation demonstrates that this approach is sound, however needs to be proven with further real world examples.

Table 23 Live goals for adaptive service compositions

Validation Set-up & Result	
Name	Live goals for adaptive service compositions.
Synopsis	This approach proposes a methodology to link requirements with the monitoring and adaptation strategies used at runtime. It proposes to enrich existing goal models with adaptive goals, responsible for the actual evolution/adaptation of the goal model at runtime. Finally, the goal model (that includes both conventional and adaptive goals) is traced onto the actual functionality provided by the system and the adaptation policies needed to make it self-adapt
Authors	Baresi, L., Pasquale, L.
Research questions	Evolution of services Continuous requirements engineering for service-based applications Context-driven adaptation based on requirements models and techniques
Scenario	The scenario is used to explain the main concepts behind the approach presented in the paper and to evaluate its viability. We apply our approach to design an application for organizing dinners out with friends. Besides booking restaurants, the application must also cope with undesired events like participants that are late or do not show up.
Method	Case study.
Description	We assessed if adaptation goals are satisfactory in representing and express adaptation at the requirements level. We demonstrated how trace conventional goals, representing the functional and non-functional requirements of the system, onto the concretes activities, partner links and variables of the process. We also demonstrated how to map adaptation goals onto process supervision rules. Finally we assessed how adaptations will be applied at runtime on the process when it is executing.
Goal	The validation is aimed to demonstrate the viability of our approach.
Set-up	Application of a case study.
Inputs	Users' requirements
Outputs	The validation demonstrates that our approach is sound and can be applied in more concrete scenarios for real applications.
Outcome	Positive (the research fulfills its goals)
Experiences	The approach has been validated on a simple toy example, and it is necessary to make more tests on real case scenarios. We also need to formalize more rigorously the way in which the process definition and its supervision rules are defined at runtime.
References	http://home.dei.polimi.it/pasquale/flags.html
Glossary	BPEL, KAOS
Keywords	Requirements, goal models, service oriented compositions

3.4 Validation Method: Questionnaire

3.4.1 Validation of the S-Cube SBA life-cycle

The development of services introduces many new types of stakeholders to the software development process - compared to the development of traditional software. Each stakeholder can take multiple roles within the lifecycle of service based applications. Table 24 documents a validation that justifies this assumption by giving a classification of stakeholders and their corresponding roles within the SBA lifecycle.

Table 24 Validation of the S-Cube SBA life-cycle through identifying stakeholders in the Service Engineering process

Validation Set-up & Result	
Name	Validation of the S-Cube SBA life-cycle through identifying stakeholders in the Service Engineering process.
Synopsis	The main output validates our primary assumption that more types of stakeholders participate in the engineering and life-cycle of SBA than with a traditional software application and that stakeholders may play different roles depending on their focus.
Authors	Gu, Q., Parkin, M., Lago, P.
Research questions	Definition of a coherent life cycle for adaptable and evolvable SBA
Scenario	N/A - is a generic piece of research on SBAs
Method	Initially, a questionnaire on Service-Oriented Software Engineering was sent all S-Cube researchers. The results were analyzed using a methodology we developed to classify and consolidate the stakeholder types. Finally, the stakeholder types found were mapped to the S-Cube life-cycle to determine their total participation in the life-cycle.
Description	The shift from monolithic application development to service provision introduces many more types of stakeholders to the software development process, each of which can take multiple roles within the lifecycle of the SBA, and who have an interest in or are influenced by the service-oriented software process. To understand these stakeholder types and roles, this work presents an initial set of stakeholder types and roles we solicited from within S-Cube. By describing these stakeholder types in the context of S-Cube's service engineering lifecycle, we demonstrate the lifecycle phases each stakeholder and role is involved in during the development and operation of SBAs. The stakeholder roles and types found and the methodology we describe for discovering them will aid the identification of the requirements for these stakeholders and contribute to future research in service engineering methodologies.
Goal	Currently, there is no common understanding of the stakeholders involved and their role(s) in the in the SBA engineering life-cycle. The goal of this work is to find these stakeholders and determine their role(s) using the collective knowledge of S-Cube researchers.
Set-up	Analysis of questionnaires, mapping of results to S-Cube SBA Engineering Life-cycle
Inputs	Questionnaire results from S-Cube researchers. Methodology for the analysis of the results.

Outputs	The main output validates our primary assumption that more types of stakeholders participate in the engineering and life-cycle of SBA than with a traditional software application and that stakeholders may play different roles depending on their focus on engineering services, SBAs, service provision or consumption, etc. Moreover, we observed that some types of stakeholders (such as *Service Architects*) are required in all of the life-cycle phases, that there is a lack of stakeholders specifically dedicated to adaptation and an absence of stakeholders playing the role of service broker. The second output is a mapping consolidated stakeholder types to the phases of the S-Cube SBA engineering life-cycle in order to determine the coverage of the life-cycle by the stakeholders. The results show the Identify Adaptation Strategies phase has the least stakeholders. This reinforces the finding of Lane, et al. (2010), which finds methods for selecting the most suitable adaptation strategy are generally not well supported.
Outcome	Positive.
Experiences	The development of Service Based Applications (SBAs) requires many more types of stakeholders than traditional software engineering and stakeholders may take multiple roles during the life-cycle of an SBA. In this result we report the research to find information about these stakeholders from institutions within S-Cube and present a taxonomy of nineteen stakeholder types performing five roles as well as a mapping between these stakeholders to the S-Cube's SBA engineering life-cycle. These observations will provide input for future research into these stakeholders that will concentrate on developing and tailoring service engineering methodologies for them.
References	Gu, Q., Lago, P., Parkin, M. (Eds): Initial Definition of User Patterns. S-Cube Deliverable CD-IA-3.1.4. 15 (2010) Lane, S., Gu, Q., Lago, P., Richardson, I.: Adaptation of service-based applications: A maintenance process? Submitted for publication (2010) Gu, Q., Parkin, M., Lago P.: A Taxonomy of Service Engineering Stakeholder Types. Submitted to the Second International Conference on Exploring Services Sciences. Geneva, Switzerland (2011)
Glossary	Life-cycle Model, Service-Based Application, Stakeholder, Service-Oriented Software Engineering
Keywords	SOA, Service-Oriented Software Engineering, Stakeholders

3.5 Validation Method: Formal Proof

3.5.1 Validation of QoS contract formation and evolution

The formation and evolution of contracts is an important aspect when defining agreements on QoS. Table 25 reports the Validation of a QoS contract formation and evolution approach. It shows how a QoS contract can be generated using a subtyping relation on the quality dimensions and value ranges.

Table 25 Validation of QoS contract formation and evolution

Validation Set-up & Result	
Name	Validation of QoS contract formation and evolution.
Synopsis	The result shows that it is possible to automatically form a QoS contract between interacting parties reusing tools we developed for managing the evolution of services. Furthermore we show that both the interacting parties can evolve under certain conditions.
Authors	Andrikopoulos, V., Fugini, M., Papazoglou, M. P., Parkin, M., Pernici, B., Siadat, S. H.
Research questions	Design, Specification & Verification of a Negotiation & Contract Agreement Protocol. Evolving Services from a Contractual Perspective.
Scenario	Automotive Purchase Order Processing was used as the basis of the validation scenario used.
Method	Formal proof, prototype
Description	We show how a QoS contract can be generated using a subtyping relation on the quality dimensions and value ranges. For that purpose, we use Allen's Interval Algebra (AIA). We also define both strict and relaxed constraints for different dimensions in order to deal with what constitutes acceptable change to different quality dimensions. In particular, we define assertion compatibility as a sufficient condition for ensuring the compatibility of provider and consumer with respect to an existing contract.
Goal	To show how QoS contracts can be automatically generated from the abstract descriptions of service providers and consumers, and under which conditions can they evolve.
Set-up	Descriptive evaluation of the findings of the formal model and proof-of-concept validation.
Inputs	A life cycle model for SLAs between service providers and consumers, a formal model for the abstract description of the QoS characteristics of service interfaces and a theory for the formation of service contracts discussed in Andrikopoulos et al. 2009.
Outputs	Definition of a partial ordering relation (subtyping) between QoS characteristics, a binding function used for contract generation, a set of different policies for the generation depending on the interpretation of the QoS constraints and a prototype proof-of-concept implementation of the proposed approach.
Outcome	Positive.
Experiences	The lack of a widely-accepted standard for describing QoS characteristics in Web services allows for flexibility in relevant research but for limited opportunities in applying such works in practice.
References	Andrikopoulos, V., Fugini, M., Papazoglou, M. P., Parkin, M., Pernici, B., Siadat, S. H.: QoS contract formation and evolution. In: Proceedings of the 11th International Conference on Electronic Commerce and Web Technologies (EC-Web 2010). (2010)
Glossary	Evolution, Quality of Service Characteristic, Service Level Agreement
Keywords	QoS contract, SLA, evolution

4 Planned validation activities

Validation activities planned for execution within a few months subsequent to the time of writing were also reported by S-Cube partners. We report one planed evaluation activity that is described in this section.

4.1 Evaluating codified user task models to discover services (CITY)

The effect of codified user task knowledge on service discovery is explored. A collection of domain-independent user task models was developed to populate an on-line catalogue accessible by one service discovery engine. The user task models encapsulate important knowledge about user goals, tasks and resources normally not available to service discovery engines. The service discovery engine matches and exploits knowledge from user task models to discover services that fit better to the matched user task. Preliminary results from a first empirical investigation indicated some refinements needed to the algorithm. The goal of the evaluation is to show that reformulation of service queries with knowledge about user tasks will increase the number of relevant services and improve the overall correctness of services retrieved by the original service discovery engine. The evaluation so far shows that codified knowledge about user tasks can be applied to improve service discovery and demonstrates its potential utility when improving the discovery of web services for an e-government SBA.

5 Conclusion

This deliverable reports on the second empirical evaluations carried out by S-Cube partners to evaluate the S-Cube research results in year three of the network. The evaluation documentation template and guidelines were outlined and the dependencies to previous deliverables and related workpackages are described. Subsequently, the documented validation results, contributed by the S-Cube partners, are presented. The validations include five different validation types, namely: prototype (14x), experiment (4x), case study (4x), questionnaire (1x), and formal proof (1x). Their distribution is shown in brackets. In total, they addressed 37 different research questions, all with a positive outcome. Subsequently, planned validation activities for the upcoming months were then presented.

References

[1] Angela Kounkou (Ed.): Results of the initial Empirical Evaluation. S-Cube Deliverable PO-IA-3.2.6, 2010.

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

[3] Kazhamiakin, R. (Ed.): Initial Definition of Validation Scenarios. S-Cube Deliverable PO-IA-3.2.1, 2009.

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

[5] Palvia, P.; Mao, E.; Salam, A.; Solman, K.: Management Information Systems Research: What's There in a Methodology? In: Communications of the Association for Information Systems, 11 (2003), p. 289–309.

[6] Easterbrook, S.; Singer, J.; Storey, M.; Damian, D.: Selecting Empirical Methods for Software Engineering Research. In: Shull, F.; Singer, J.; Sjøberg, D. I. K. (Eds.): Guide to Advanced Empirical Software Engineering. Springer, London, 2007, p. 285–311.

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

[8] Sonja Zaplata, Christian P. Kunze and Winfried Lamersdorf. Context-based Cooperation in Mobile Business Environments: Managing the Distributed Execution of Mobile Processes. In Business and Information Systems Engineering (BISE), Vol. 2009(4), 2009.

Appendix

Table 26 Migratable XPDL processes and comparison to process fragmentation

	Tested XPDL elements	Process migration	Process fragmentation
Atomic activities	*activity*	possible	possible
Structured activities	*activity set* branches (*XOR*) loops branches (*AND*)	possible possible possible replication and synchronization required	possible possible (transfer of decision) coordination required coordination and synchronization required
Other elements	*transaction (XPDL 2.0)*	for *undo*: to be avoided for *compensation*: possible	coordination required
Dead path elimination	-	automatically	coordination required
Privacy of process parts	-	artificial	automatically
Splitting atomic activities	-	forbidden	no known approach
Data replication	-	only for parallel execution (entire process)	always: *data fields, data types, applications, participants*
Design time distribution	-	possible (assign all activities in advance)	possible
Runtime distribution	-	during execution	once after invocation

Table 27 Migratable WS-BPEL processes and comparison to process fragmentation

	Tested WS-BPEL elements	Process migration	Process fragmentation
Atomic activities	*invoke* *reply* *receive* *assign* *wait, empty, exit* *throw, rethrow*	possible possible (log) fixed participant possible possible possible (log)	possible coordination required fixed participant possible possible coordination required
Structured activities	*sequence* *if then else* *while, repeat until, for each* *pick* *flow*	possible possible possible possible, but small risk of missing events coordination required	possible unnecessary fragments coordination required potential replication of events and/or additional coordination coordination required
Other elements	*scope* *fault handler* *compensation handler*	generally available generally available generally available	coordination required coordination required coordination required
Dead path elimination	-	automatically	coordination required
Privacy of process parts	-	artificial	automatically
Splitting atomic activities	-	forbidden	no known approach
Data replication	-	only for parallel execution (entire process)	always: *variables, scopes,* optionally: *events*
Design time distribution	-	possible (assign all activities in advance)	possible (equivalent to service choreography)
Runtime distribution	-	during execution	once after invocation