Grant Agreement N° 215483

Title:	*Integrated principles, techniques and methodologies for specifying end-to-end quality and negotiating SLAs and for assuring end-to-end quality provision and SLA conformance*
Authors:	*UniDue, TUW, CITY, Polimi, INRIA, Lero, USTUTT, UPM, SZTAKI, UCBL, FBK, UoC, UPC, SEERC*
Editor:	*Osama Sammodi and Andreas Metzger (UniDue)*
Reviewers:	*Clarissa Marquezan (UniDue), Michael Parkin (Tiburg) and Xavier Franch (UPC)*
Identifier:	*CD-JRA-1.3.5*
Type:	*Contractual Deliverable (CD)*
Version:	*1.0*
Date:	*March 19, 2011*
Status:	*Final*
Class:	*External*

Management summary

The aim of this deliverable is twofold: (1) It provides an updated overview of the research challenges of WP-JRA-1.3 ("End-to-End Quality Provision & SLA Conformance"). (2) It reports on results for run-time quality assurance, quality prediction (to enable proactive adaptation) and automated and proactive negotiation, where first improvements based on the validation results are incorporated. Work related to these principles and techniques, carried out by S-Cube NoE participants and published in books, journals and conference proceedings, is summarized and assessed with respect to the coverage of the research challenges for this workpackage.

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) – UniDue	Germany
Tilburg University – Tilburg	Netherlands
City University London – CITY	U.K.
Consiglio Nazionale delle Ricerche – CNR	Italy
Center for Scientific and Technological Research – FBK	Italy
French Natl Institute for Research in Computer Science and Control – INRIA	France
The Irish Software Engineering Research Centre – Lero	Ireland
Politecnico di Milano – Polimi	Italy
MTA SZTAKI – Computer and Automation Research Institute – SZTAKI	Hungary
Vienna University of Technology – TUW	Austria
Universit Claude Bernard Lyon – UCBL	France
University of Crete – UOC	Greece
Technical University of Madrid – UPM	Spain
University of Stuttgart – USTUTT	Germany
University of Amsterdam – VUA	Netherlands
University of Hamburg – UniHH	Germany

Associate Members of the S-Cube consortium:

Centre of Excellence in Information and Communication Technologies – CETIC	Belgium
Centre for Service Research, Manchester Business School, The University of Manchester	U.K.
Dipartimento di Informatica, Università di Pisa	Italy
Faculty of Informatics, Università della Svizzera Italiana – USI	Switzerland
Institute of Computer Science, University of Münster	Germany
IT and Media Center & Department of Computer Science, Dortmund University of Technology	Germany
IT Innovation Centre, University of Southampton	U.K.
Karlsruhe Institute of Technology – KIT	Germany
School of Software, Tsinghua University	Peoples' Republic of China
Semantic Technology Institute Innsbruck, University of Innsbruck	Austria
Service & Information Systems Engineering Department, Universitat Politècnica de Catalunya – UPC	Spain
SINTEF	Norway
South East European Research Centre – SEERC	Greece
University of L'Aquila	Italy
Université Européenne de Bretagne / Université de Bretagne-Sud	France
University of Groningen	Netherlands
Universidad de Sevilla	Spain

Published S-Cube documents

These documents are all available from the S-Cube Web Portal at `http://www.s-cube-network.eu/`

Foreword

Workpackage JRA-1.3 of S-Cube ("End-to-End Quality Provision & SLA Conformance") has been designed to achieve four long-term objectives:

1. To define principles, techniques and methodologies for *specifying, negotiating and assuring end-to-end quality provision and SLA conformance (including proactiveness)* with respect to quality characteristics across the functional layers for service infrastructure, service composition and coordination, and business process management, and, across the chain of service providers and consumers. The quality characteristics considered will include, but will not be limited to, performance, dependability, reliability, availability, usability and accessibility.

2. To specify clearly defined interfaces and the interrelationships with respect to end-to-end quality aspects:
   - Between functional layers service infrastructure, service composition and coordination, business process management and the SBA engineering framework, and
   - Between the SBA engineering framework and the SBA monitoring and adaptation framework.

3. To shape the S-Cube convergence knowledge model by providing an integrated set of principles, techniques and methodologies for end-to-end quality assurance and SLA conformance.

4. To provide contributions to IA-3, where the results are integrated into the S-Cube Framework for Service-Based Applications.

As part of these long-term goals, this deliverable provides: (1) an updated overview of the research challenges being pursued within WP-JRA-1.3, and (2) a report on results for run-time quality assurance, quality prediction (to enable proactive adaptation) and automated and proactive negotiation, where first improvements based on the validation results are incorporated, thereby contributing to the research challenges of the WP.

Acknowledgments: The editors would like to thank the authors of the papers, technical reports, articles and book chapters described in this deliverable for allowing their work to be used in this document.

Contents

Chapter 1

Deliverable Overview

1.1 Introduction

Service-orientation is increasingly adopted as a paradigm for building highly dynamic, distributed and (self-)adaptive software systems, called service-based (or service-oriented) applications (SBAs).

An SBA is realized by composing individual software services. In contrast to a software component, for the service consumer (or service composer) a software service is not an individual piece of software. Instead, the service consumer can only access the functionality and quality provided by that piece of software via the service's interface. There is a clear trend that in the future SBAs will increasingly be composed from third-party services that are accessible over the Internet [56]. SBAs based on third-party services allow taking the concept of ownership to the extreme: not only is the development, quality assurance, and maintenance of the software under the control of third parties, but the software itself is also operated and managed by them [12].

This scenario implies a fundamental change to how software is developed, deployed, and maintained [12]. An SBA cannot be specified and realized completely in advance (i.e, during design-time) due to the incomplete knowledge about the third-party services as well as the system's context and communication infrastructure [8]. Thus, compared to traditional software systems, much more decisions need to be taken during the operation of the SBA (i.e., after it has been deployed), once the missing knowledge is available.

To provide the desired end-to-end quality of such distributed SBAs, the dynamic agreement and assurance of service quality becomes an important issue. This requires that not only quality aspects are negotiated and agreed between the service providers and the service requestors (also known as service consumers), but also that quality aspects are checked during run-time (i.e., the operation of the SBA) to determine – or even predict – whether there is a need for adapting the SBA or for re-negotiating the quality contracts.

Three main activities relevant for quality within SBAs can be identified based on the general life-cycle of electronic contracts [50, 60]:

- *Quality definition*: This concerns the definition of a model or language for the specification of contract terms, which is understood and shared by the (contracting) parties. This model or language then is used to instantiate an actual contract (e.g., a SLA) that reflects the domain dependent interests of providers and consumers, or to state the end-to-end quality requirements towards an SBA.
- *Quality negotiation*: Establishment of an electronic contract concerns the set of tasks that is required for defining actual contracts. This may involve the selection of service providers (the contract partners) among a set of potential providers, the negotiation of the contract terms between the selected providers and the service consumer, and the agreement to the contract terms.

- *Quality assurance*: This concerns tasks for assuring the satisfaction of the contracts and the fulfilment of the expected end-to-end requirement. In the case of quality contracts, this implies assuring that the quality levels negotiated and agreed between the service provider and the service requestor are met.

This deliverable introduces novel contributions for the above activities, specifically focusing on predicting the future quality of SBAs (see Section: 2.1.3.2).

1.2 Deliverable Structure

As a paper-based deliverable, this deliverable contains two parts: (1) This document, which provides the overall motivation and overview of the key research outcomes of the workpackage. (2) The actual research publications that describe the workpackage outcomes in detail and which are summarized in this document.

The document, is structured as follows: In Section 2, this deliverable provides an updated overview of the research challenges being pursued within WP-JRA-1.3. In Section 3, the document – based on research publications of the workpackage members and the associate members of S-Cube – reports on a set of principles and techniques for quality definition, negotiation and assurance developed in year 3 of the network, thereby contributing to the research challenges of the workpackage. Section 4 concludes the deliverable and provides an outlook to future work in the workpackage.

Chapter 2

Research Challenges and Contribution to the Integrated Research Framework

As described in S-Cube's Description of Work (DoW), the general research goal of workpackage WP-JRA-1.3 is to devise novel principles, techniques and methods for defining, negotiating and assuring end-to-end quality across the functional layers as well as across networks of service providers and consumers.

This chapter provides an update of the research vision for this workpackage by defining and refining the research challenges addressed in WP-JRA-1.3, which address key research gaps identified in [47]. Thereby, this chapter allows for relating the results presented in this deliverable to the research objectives of the workpackage (see Sections 2.2 and 3). Specifically for year 3 of the project, focus of research activities was on "proactiveness" (cf. Section 2.1.2.2 and 2.1.3.2).

2.1 Key Research Challenges

Figure 2.1 provides and overview of the WP's research challenges in the context of the quality activities (as introduced in Section 1.1). Those research challenges are detailed below.

2.1.1 Challenges In Quality Definition

2.1.1.1 End-to-End Quality Reference Model

Motivation: Different kinds of quality attributes are important in an SBA. There is thus a strong need for methods that address quality attributes in a comprehensive and cross-cutting fashion across all layers of an SBA. Due to the dynamism of the world in which SBAs operate, techniques are needed to aggregate individual quality levels of the services involved in a service composition in order to determine and thus check the end-to-end quality during run-time of the application. This aggregation will typically span different layers of an SBA and thus a common understanding of what the different quality attributes mean within and across these layers is needed.

Challenge: To support end-to-end quality provision, S-Cube has aimed at making the dependencies between different kinds of quality attributes explicit. For instance, the interrelation between the fulfillment of different QoS attributes across the various layers has been modeled. In addition, S-Cube has aimed at understanding the dependencies between quality of information (QoI) attributes on the infrastructure layer, the satisfaction of quality of experience (QoE) on the service composition layer and the achievement of quality of business (QoBiz; business value or business KPIs). One key means to achieve the above objective has been to achieve a shared understanding of quality attributes between the S-Cube layers and disciplines by defining the S-Cube Quality Reference Model. Based on the S-Cube Quality Reference Model and the quality definition language (see Challenge "Rich and Extensible Quality

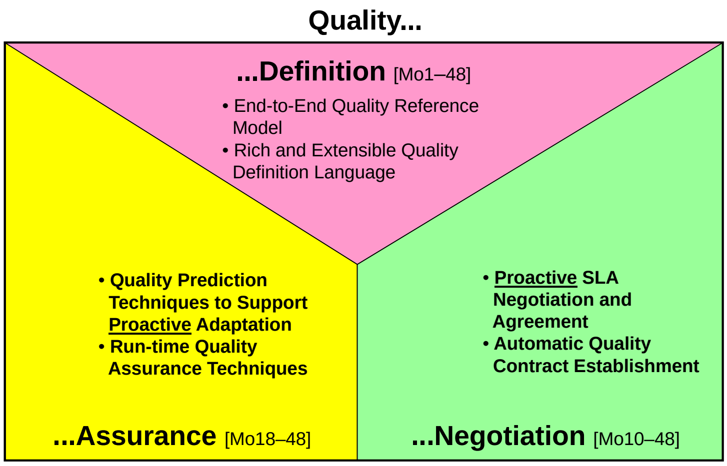

Figure 2.1: Activities relevant for quality of SBAs and key research objectives of workpackage

Definition Language" in Section 2.1.1.2 below), foundations for techniques will be devised, which allow aggregating individual quality levels of the services involved in a service composition in order to determine and thus ultimately check end-to-end quality.

Note: Work on this challenge constituted the foundations for the work on other challenges of the WP. Those activities have been finalized – as planned – by month 12 of the project.

2.1.1.2 Rich and Extensible Quality Definition Language

Motivation: Concerning quality modeling and definition, this project has observed that there is a lack of an established, standardized, rich, extensible and semantically well-defined quality definition language [51]. As a result, quality capabilities and requirements, as well as service SLAs are described by many different formalisms and languages, such as the WSLA language [35], WSML [52], SLAng [29] and RBSLA [49] (amongst many others). Due to this fragmentation, there is still a requirement for a standardized and definition language — necessary for interoperable services.

Challenge: S-Cube has developed the concepts for a quality definition language (i.e., the S-Cube Quality Meta Model), which allow describing every relevant aspect of quality for services and SBAs, including metrics, units, measurement functions and directives, constraints, value types, etc. In addition, this quality definition language encompasses a rich set of domain-dependent and global quality attributes and is extensible so as to allow the addition of new quality dimensions when it is needed (e.g., for an application domain which has currently not been considered). As a starting point, the set of quality attributes as defined in the S-Cube Quality Reference Model (see Challenge "End-to-End Quality Reference Model" above) has been exploited. Further, this standard quality definition language is semantically enriched - where feasible - to be machine-processable or machine-interpretable. This quality definition language is created to be applicable in complex SBAs, in which services can be invoked and composed with variable quality profiles. The quality definition language is capable of expressing quality capabilities and SLAs by using functions, operators and comparison predicates on quality metrics. It also allows the description of composition rules for possible combinations of composition constructs and quality metrics.

Note: Similar to the challenge 2.1.1.1, the quality definition language serves as foundation for work on the other challenges and thus has been closed by M25.

2.1.2 Challenges In Quality Negotiation

2.1.2.1 Automatic quality contract establishment

Motivation: Service negotiation and agreement involves selecting one out of many service providers based on his quality offer so as to agree on and thus establish the contracts for the delivered service. To address dynamic adaptations of SBAs, a growing need for automating the negotiation and agreement of quality attributes (e.g., as stipulated by SLAs) can be observed. However, this issue requires considering user interaction and experience (e.g., QoE) issues that may impact on the negotiation itself. This aspect requires a multi-disciplinary effort in which technology researchers will have to interact with researchers addressing user interaction issues.

Challenge: One key research objective regarding quality contract establishment is to exploit user and task models, which codify user preferences and characteristics (see JRA-1.1), in order to devise advanced automated negotiation techniques and protocols. Those advanced techniques could lead to service negotiators (e.g., autonomous components provided as core services) that perform the negotiation process on behalf of the service consumers (requestors) and providers.

2.1.2.2 Proactive SLA negotiation and agreement

Motivation: Similar to proactive (and possibly automated) adaptation (see Challenge "Quality Prediction Techniques to Support Proactive Adaptation" in Section 2.1.3.2), proactive SLA negotiation and agreement is a key prerequisite for effective run-time SLA negotiation since negotiation may have a significant computational cost and, therefore, undertaking it when there is an immediate need to use a new service can be unlikely or unfeasible at run-time.

Challenge: The challenge for quality contract negotiation and agreement is how to negotiate the terms and conditions under which a service can be offered before the need for deploying or invoking these services arises. Many of these challenges lie in the definition of negotiation models, and to make the envisioned advances in automated negotiation. We aim to address the limitations introduced above by starting negotiation when there is evidence that the need for deploying a new service and/or change the conditions of deploying a current service is likely to arise but has not arisen yet. Thus, our proactive negotiation approach is based on forecasting at run-time a number of factors related to the deployment of services. Those include, for example, the expected demand for a service, the expected levels of service provision, and the expected service terms and conditions that a service negotiator is likely to agree. The availability of accurate forecasts can lead to effective proactive run-time negotiation strategies for service clients. Prediction also plays a role in quality prediction for proactive adaptation (see Challenge "Quality Prediction Techniques to Support Proactive Adaptation" in Section 2.1.3.2). Although the factors which are relevant differ in both situations, we expect to be able to exploit synergies between the principles and techniques that are developed.

2.1.3 Challenges in Quality Assurance

2.1.3.1 Run-time Quality Assurance Techniques

Motivation: Given the need for adapting SBAs at run-time, quality assurance techniques that can be applied at run-time are essential. The major type of run-time quality assurance techniques used today is monitoring, which is often classified as passive (when monitoring relies on actual inbound service consumer traffic to take measurements, so problems can only be discovered after they have occurred) or active (e.g., during run-time testing where the consumer traffic is generated by the testing agent).

Monitoring observes the SBA (or its constituent services) during their current execution, i.e., during their actual use or operation. However, monitoring only allows the assessment of the quality of 'representative' applications (in fact the application in operation) and thus key problems might only be discovered by coincidence. In contrast, standard and consolidated software quality assurance techniques employed during design time, can uncover problems that might only occur after many invocations of the SBA. As an example, model analysis can examine classes of executions, thereby leading to more universal statements about the properties of the artifacts.

Challenge: S-Cube investigates in how standard and consolidated offline software quality assurance techniques can be extended to be applicable while the application operates. For instance, we investigate into run-time model analysis techniques and other online techniques such as online testing. In addition to extending the quality assurance techniques to the operation phase, synergies between the different classes of quality assurance techniques are exploited. As an example, we investigate how testing can be combined with monitoring in such a way that when a deviation is observed during monitoring, dedicated test cases are executed in order to determine - with high confidence - the cause for the deviation. In order to achieve feasible results from run-time quality assurance, it is essential that the artifacts exploited for run-time analysis or testing are a consistent and up-to-date representation (abstraction) of the running SBA. For example, this leads to the challenge on how to 'synchronize' the model with the SBA in operation in order to achieve valid analysis results. Existing quality assurance techniques appear to be not yet fully incorporated into a comprehensive life-cycle. These aspects are particularly critical as the designers find it quite difficult to understand what will happen as a result of some self-adaptation design choice. Research, jointly with WP-JRA-1.1, thus addresses the consistent and comprehensive integration of quality assurance into the service life-cycle (see JRA-1.1).

2.1.3.2 Quality Prediction Techniques to Support Proactive Adaptation

Motivation: To respond in a timely fashion to changes implied by the highly dynamic and flexible contexts of future SBAs and to promptly compensate for deviations in functionality or quality, SBAs have to be able to self-adapt. In current implementations of SBAs, monitoring events trigger the adaptation of an application. Thus self-adaptation often happens after a change or a deviation has occurred. Yet, such reactive adaptations have several drawbacks, such as: (1) Executing faulty services can lead to unsatisfied users and typically requires the execution of additional activities (e.g., compensation or roll-back); (2) Execution of adaptation activities takes time and thereby can reduce the system performance; (3) It can take time before problems in the system lead to monitoring events (e.g., time needed for the propagation of events from the infrastructure to the business process level), thus events might arrive so late that an adaptation of the system is not possible anymore (e.g., because the system is in a deadlock situation).

Proactive adaptation presents a solution to address these drawbacks, because - ideally - the system will detect the need for adaptation and will self-adapt before a deviation will occur during the actual operation of the SBA and before such a deviation can lead to the above problems. Key to proactive adaptation is to predict the future quality (and functionality) of a SBA and to proactively respond if the prediction uncovers deviations from expected quality (or functionality).

Challenge: To support the vision of proactive adaptation, S-Cube works on devising novel quality prediction techniques. Depending on the kind of quality attribute to be predicted, these can range from ones that are built on traditional techniques (see Challenge "Run-time Quality Assurance Techniques" in Section 2.1.3.1) to ones that exploit modern technologies of the Future Internet. As an example for the first case, correctness or performance (QoS) could be predicted by building on techniques similar to online testing, run-time model analysis or model-checking at run-time. As an example for the latter case, usability of services (QoE) could be predicted by extending existing principles of reputation systems. In this setting, quality prediction approaches based on data mining appear promising.

2.2 Contribution of Deliverable to Key Research Challenges

Summarizing the above challenges, WP-JRA-1.3 pursues integrative and innovative research to devise novel principles, techniques and methods for defining, negotiating and assuring end-to-end quality for SBAs.

The aim of this deliverable is to present the outcomes on the integrated and joint research on quality assurance for year 3 of the project. More specifically, this document summarizes the key research outcomes in addressing challenges in Quality Negotiation (see Section 2.1.2) and Quality Assurance (see Section 2.1.3). In the remainder of this document, those research outcomes will be presented and related to the WP challenges as well as to the other S-Cube WPs in more detail.

Chapter 3

Principles, Techniques and Methodologies for Service Quality

The main research results in the context of this deliverable have been published or submitted as research papers, articles and book chapters. Thus, in this section we will present (in compact form) the contributions of those papers and how these results relate to the WP research challenges described earlier in Section 2.1.

It should be noted that due to S-Cube being a Network of Excellence, and not an Integrated Project, the papers that constitute this deliverable present solutions to the WP challenges from different angles rather than different "views" on the very same technical solution. More important for S-Cube is the fact that those papers document a significant step towards integration of the different research communities that participate in S-Cube. As an example, the techniques exploit software engineering solutions (such as testing, verification or model-driven development) and techniques from SOC (service composition) and service infrastructures to address problems specific to SBAs.

3.1 Structured Presentation of Results

The following 8-part structure, inspired in parts by "How to Get Your Paper Accepted at OOPSLA" [45], is used to describe each of the reports, papers and articles that form the results of this deliverable.

- *Context and Background:* Initially, the context and background of the problem being addressed in the paper is provided.
- *Problem Statement:* Based on the background, the problem that is addressed (i.e., the research question which is answered) is motivated and explained.
- *Relevance of the Problem and Progress from State of the Art:* The explanation on why the problem is relevant is important to understand why the problem (i.e., research question) is worth pursuing. In addition, the relation of the work to the state of the art helps understanding the novelty of the contribution and its progress from existing work.
- *Relation to WP Challenges:* The contribution to the WP research challenges is described to understand the contribution of the paper to the overall aims of the deliverable (cf. Section 2.2) and the WP.
- *Solution / Research Method:* Either the (innovative) solution (idea) to the problem is stated or the employed research method (e.g., empirical study) is described.
- *Benefits and Evaluation:* The benefits and utility of the solution when applied to the problem is stated, and, if applicable, it is described how those benefits have been demonstrated by means of an evaluation (method of evaluation and results).
- *Relation to Research Framework:* The solution of the paper is related to the elements of the S-Cube Research Framework and thus to S-Cube JRA work packages, thereby describing the integration

achieved across JRA-1 and JRA-2: Monitoring and Adaptation (JRA-1.2), Engineering and Design (JRA-1.1), BPM (JRA-2.1), Service Composition and Coordination (JRA-2.2), and Service Infrastructure (JRA-2.3).

- *Discussion and Future Work:* Critical discussion on what are the current gaps and shortcomings of the solution and which future research activities are planned. This will allow shaping the future research roadmap for the WP.

3.2 Summary of Research Results

Table 3.2 summarizes the research results (i.e., papers) by categorizing them in relation to their contribution to the six research challenges for this workpackage, which were described earlier in Sections 2.1.1.1–2.1.3.2. As the table shows, the majority of the work presented here concentrates on the quality assurance challenges, including run-time quality assurance techniques and quality prediction techniques to support proactive adaptation and quality negotiation (especially proactive negotiation). This is in-line with the focus of this deliverable, which is on reporting on results for run-time quality assurance, quality prediction (to enable proactive adaptation) and automated and proactive negotiation, where first improvements based on the validation results are incorporated, thereby contributing to the research challenges of the WP. The research results of S-Cube concentrating on the quality reference model and on the quality definition language research challenges can be found in the S-Cube deliverables CD-JRA-1.3.2 [18] and CD-JRA-1.3.3 [27], respectively.

The table also shows the integration of this work with the other workpackages of the S-Cube research framework. What is interesting is the number of research results also contributing to JRA-2 (the realization mechanisms for SBAs) and to the research taking place in WP-JRA-2.2 into adaptable coordinated service compositions as well as JRA-2.3 (adaptation of infrastructure). Most of the results also contribute to workpackage WP-JRA-1.2, adaptation and monitoring principles, techniques and methodologies for SBAs. Finally, some of the results contribute to workpackage JRA-1.1, engineering principles, techniques and methodologies for hybrid SBAs, and to JRA-2.1, BPM.

Sections 3.2.1–3.2.19 now provide the descriptions of the research results according to the standard structure described above.

3.2.1 "Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution" [32]

Context and Background: In service-oriented computing, service consumers and service providers agree on QoS properties of services in service level agreements (SLAs). SLAs specify Service Level Objectives (SLOs), which define numerical target values on metrics such as response time or availability of a service (e.g., maximum response time is 30 seconds) which the provider is obliged to provide. These metrics are monitored at runtime in order to check SLA conformance. SLO Violations negatively impact consumer satisfaction and often lead to penalty payments. Therefore, the service provider wants to be aware of SLA violations and wants be able react to them in a timely fashion.

Problem Statement: In [33] and deliverable JRA-1.3.4 we have described an approach which enables prediction of SLA violations in service compositions. That approach is useful in that it alerts the service provider of potential violations, however it does not yet support the prevention of violations. In that respect, an approach is needed which allows adapting the service compositions before SLA violations occur. In the context of service compositions there are several potential adaptation mechanisms one could use, such as service substitution, fragment substitution, and infrastructural reconfiguration. In this paper, we focus on adaptation via composition fragment substitution.

Section	Paper Title [reference]		Quality Definition		Quality Negotiation		Quality Assurance		
			Quality Reference Model	Quality Definition Language	Automatic Contract Establishment	Proactive SLA Negotiation & Agreement	Run-time Quality Assurance	Quality Prediction for Proactive Adaptation	Also Contributes to Workpackage
3.2.1	(+) Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution	[32]					✓	✓	JRA-1.2, JRA-2.2
3.2.2	(+) Monitoring, Prediction and Prevention of SLA Violations in Composite Services	[31]					✓	✓	JRA-2.2, JRA-2.3
3.2.3	(+) Semantic Resource Allocation with Historical Data Based Predictions	[15]				✓	✓	✓	JRA-1.2, JRA-2.3
3.2.4	(+) Building Dynamic Models of Service Compositions With Simulation of Provision Resources	[25]			✓			✓	JRA-2.2
3.2.5	(+) QoS Assurance for Service-Based Applications Using Discrete-Event Simulation	[26]					✓	✓	JRA-2.2
3.2.6	(+) Enabling Proactive Adaptation through Just-in-time Testing of Conversational Services	[13]					✓	✓	JRA-1.1, JRA-1.2
3.2.7	(*) Usage-based Online Testing for Proactive Adaptation of Service-based Applications	[53]					✓	✓	JRA-1.2, JRA-2.2
3.2.8	(o) Towards Proactive Adaptation with Confidence: Augmenting Monitoring with Testing	[38]					✓	✓	JRA-1.2, JRA-2.2
3.2.9	(+) Towards Proactive Adaptation: A Journey along the S-Cube Service Life-Cycle	[39]				✓	✓	✓	JRA-1.2, JRA-2.2. JRA-1.1
3.2.10	(+) A Soft Constraint-Based Approach to QoS-Aware Service Selection	[63]						✓	JRA-1.2, JRA-2.2
3.2.11	(+) Towards data-aware qos-driven adaptation for service orchestrations	[24]					✓	✓	JRA-1.2, JRA-2.2
3.2.12	(+) Test Coverage of Data-Centric Dynamic Compositions in Service-Based Systems	[23]					✓		JRA-2.2
3.2.13	(+) A Framework for Proactive SLA Negotiation, Proactive SLA Negotiation for Service Based Systems	[36, 37]				✓			JRA-1.2
3.2.14	(*) A Survey on Service Quality Description	[28]	✓	✓					JRA-1.1
3.2.15	(+) A dynamic privacy model for web services	[40]				✓			JRA-1.2
3.2.16	(+) Enhancing Service Network Analysis and Service Selection using Requirements-based Service Discovery	[62]	✓		✓		✓		JRA-2.1, JRA-1.1
3.2.17	(+) A semantic based framework for supporting negotiation in Service Oriented Architectures	[10]			✓	✓			JRA-1.2, JRA-2.3
3.2.18	(o) High-quality Business Processes based on Multi-dimensional QoS	[34]				✓	✓	✓	JRA-1.2, JRA-2.2
3.2.19	(o) A CMMI Based Configuration Management Framework to Manage the Quality of Service-Based Applications	[21]					✓		JRA-1.2, JRA-2.2, JRA-1.1

Table 3.1: Coverage of Research Challenges by Research Results ((*) = submitted, (+) = accepted), (o) = submitted in JRA-1.3.4 and now published

Relevance of Problem and Progress from State of the Art: The work presented in this paper is an extension of our previous approaches described in [33, 31]. In [33] we deal only with the prediction aspect, while in [31] prevention based on prediction is realized via service substitution.

In this work, we extend those two approaches by enabling a more sophisticated adaptation mechanism, namely the substitution of whole composition fragments in order to prevent SLA violations.

Relation to WP Challenges: The paper addresses the JRA-1.3 research challenge of "Quality Prediction Techniques to Support Proactive Adaptation" and partially "Run-time Quality Assurance Techniques", as the prediction occurs during run-time. It also addresses the JRA-1.2 research challenge "Proactive Adaptation and Predictive Monitoring" and the JRA-2.2 challenge "QoS Aware Adaptation of Service Compositions" as the approach focuses on service compositions.

Solution / Research Method: In this paper we show how the application of the aspect-oriented programming paradigm to runtime adaptation of service compositions can be used to prevent SLA violations. Adaptations are triggered by predicted violations as described in [33], i.e., at certain checkpoints in the process the SLO value is predicted using regression models which have been trained based on past process instances. The adaptations are implemented as substitutions of fragments in the service composition. Composition fragments are modeled as separate entities implementing a part of business logic, and are linked into the original composition via a special type of activities, so called virtual activities. After a violation is predicted, we evaluate available alternative composition fragments with respect to their expected impact on the performance of the composition, and choose those fragments which are best suited to prevent a predicted violation. The selected fragment is woven into the running composition instance using AOP techniques.

Benefits and Evaluation: The paper presents an architecture of the approach and a prototype implementation which is based on the Windows Workflow Foundation technology. Using a purchase order processing example run on our prototype, an experimental evaluation has been conducted, which shows that our approach is able to prevent predicted SLO violations effectively and with a small performance overhead.

Relation to Research Framework: The approach, from a mechanisms point of view, focuses on the Service Composition and Coordination layer (JRA-2.2), as service compositions and their fragments are monitored and adapted. In addition, the approach is relevant for Monitoring and Adaptation (JRA-1.2) and Quality Assurance and Prediction (JRA-1.3).

Discussion and Future Work: In future work, we will extend our approach in several directions. Firstly, we will take into account that adaptations which improve one SLA metric can easily lead to a violation of another SLA metric. This is in particular true for cost. Therefore, the impact model will be extended. Secondly, currently we assume that the number of available alternative fragments is small so that finding the best combination by fully enumerating all possibilities is feasible. In future work we will work on heuristic optimization for cases where full enumeration is not feasible.

3.2.2 "Monitoring, Prediction and Prevention of SLA Violations in Composite Services" [31]

Context and Background: Nowadays, many companies are shifting towards the Software-as-a-Service (SaaS) model, providing coarse-grained valued-added services as a composition of existing Web Services. In this context the concept of Service Level Agreements (SLAs), which are contracts between providers of composite services and their customers, is very important. SLAs govern the quality that customers can expect from the service. Quality requirements are formulated as a collection of Service

Level Objectives (SLOs), which are numerical target values, and penalties for not fulfilling these objectives. For the service provider it is therefore vital to minimize cases of SLA violation (i.e., cases where one or more objective could not be fulfilled).

Problem Statement: Currently, most research focuses on finding the reason for SLA violations *ex post* (after the violation has happened). While this is certainly useful for later analysis and improving the composition it does not directly help preventing violations (i.e., the damage has already been done). In this work we present an *ex ante* approach, which allows for runtime prevention of violations, before they have happened.

Relevance of Problem and Progress from State of the Art: This paper is an continuation of our work presented in [33]. We build means to automatically adapt service compositions on top of our earlier work on prediction of violations. The work presented in this paper can be seen as part of SLA management. SLA management includes the definition, deployment and monitoring of SLAs. In our work, we add another aspect to this, namely the prediction and automated prevention of SLA violations from the provider's point of view before they have actually occurred. We assume that the service has been implemented as a service composition. Hence, adaptation boils down to applying one or many adaptation actions to the service composition. These actions can be on many three levels: *Data Manipulation* adaptations are the most simple type of adaptation action. In this case the service composition is in fact not changed. Instead the data flow of the composition is intercepted and adapted. More complex than data manipulation is *Service Rebinding*. We currently support all kinds of 1:1 rebinding, i.e., all cases where one service invocation is mapped to exactly one different invocation. Finally, it is also possible to adapt the structure of the composition. This more complex type of adaptation has been published as part of a separate research paper [32].

Relation to WP Challenges: The paper addresses the WP challenge "Run-time Quality Assurance Techniques" and "Quality Prediction Techniques to Support Proactive Adaptation". Furthermore, the paper has a strong relationship to the research conducted in WP-JRA-2.2 (adaptable service composition) and WP-JRA-2.3 (service infrastructures for adaptable service compositions).

Solution / Research Method: The paper presents solutions to monitoring, prediction and prevention of SLA violations in service compositions. These three integrated components are discussed based on an illustrative example (an assembling service). Our discussion is based on an implementation of the system within the larger VRESCo research prototype [42]. Furthermore, the paper shows the validity of the approach based on an implementation of the case study and numerical validation.

Benefits and Evaluation: The evaluation in the paper clearly shows the usefulness of our approach. We show that, based on which adaptation strategy is used, 60% to 78% of all predicted SLA violations can be prevented. Of course this concrete number is case-specific, and other cases might lead to significantly higher or lower prevention rates. However, the experiment still shows that the approach can in general be beneficial, under the assumption that there are actually adaptation actions available that are able to prevent the SLA violations in the first place.

Relation to Research Framework: As stated above, the paper is relevant for the work packages WP-JRA-1.3 (adaptation for SLA compliance), WP-JRA-2.2 (adaptable service compositions) and WP-JRA-2.3 (infrastructure support for adaptable service compositions, and service monitoring). The work conducted in this paper will feed back into the research in all of these three work packages.

Discussion and Future Work: Preventing violations of SLAs is a main concern for providers of service compositions aiming at satisfying their customers. While most current research in the area considers the explanation of violations after they have happened we propose a framework for runtime prediction and subsequent prevention of violations. Our system is based on the idea of monitoring and analyzing runtime data to trigger adaptation actions in endangered composition instances. Our system is based on the VRESCo runtime environment, which provides facilities used for monitoring and adaptation. We have shown how our framework can be successfully used to significantly reduce the number of violations in an illustrative case study.

While the current version of our work is promising, there are still some open issues. Most importantly our current approach does not scale to a larger number of adaptation actions per checkpoint, since optimization is based on full enumeration. Secondly, the adaptation actions supported so far are somewhat limited in that complex structural adaptations of the composition are not supported. Finally, we currently do not take the costs of adaptations into account (e.g., there are costs associated with using express shipping instead of regular shipping, which may in some cases be higher than the gain of not violation the SLA). We plan to work on all of these issues as part of our future work in the S-Cube project.

3.2.3 "Semantic Resource Allocation with Historical Data Based Predictions" [15]

Context and Background: One of the most important issues for Service Providers in Cloud Computing is delivering a good quality of service. This is achieved by means of the adaptation to a changing environment where different failures can occur during the execution of different services and tasks. Some of these failures can be predicted taking into account the information obtained from previous executions. The results of these predictions will help the schedulers to improve the allocation of resources to the different tasks.

Problem Statement: The Service-Oriented Computing (SOC) paradigm relies on the composition of services to build distributed applications by using basic services offered by third parties. From the business point of view, the service provider agrees with its customers the Quality of Service (QoS) and level of service through a Service Level Agreement (SLA). The fulfilment or violation of the SLAs indicate the grade of satisfaction of customers with the SP, affecting directly or indirectly the revenue of the providers. One of the most common SLA violations happens when unexpected events such as failures appear and the system is not able to adapt to this change. Therefore, the goal of service providers is to minimize the effects of SLA violations, for which they can utilize techniques to detect and predict violations and to adapt running processes in order to avoid or decrease the bad effects of violations. The current paper provides a practical solution for using prediction and adaptation techniques jointly in the service provider's infrastructure. We introduce a generic framework for prediction and adaptation, and describe its application in a concrete scenario (Cloud resource scheduling).

Relevance of Problem and Progress from State of the Art: In the paper we argue for establishing a full loop of control for service executions including monitoring, collection and evaluation of monitoring data, identification of adaptation needs and finally adapting running processes. We provide a working solution for such a complete cycle. Earlier solutions implement specific, static allocation policies for solving this problem, which cannot be changed easily. In our work we have introduced the multi-agent solution in the Semantic Resource Allocation process leaving users the availability of extending or changing the policies. In our system, customers and providers can describe the scheduling rules that are the most convenient for their interests. Those policies will be combined during the negotiation, trying to get a solution which satisfies all policies. While other papers on prediction techniques exist, the seamless combination of using semantic prediction data and automatic yet customizable adaptation based on prediction data gives the uniqueness of our approach.

Relation to WP Challenges: Regarding research challenges the paper is mostly related to "Quality Prediction Techniques to Support Proactive Adaptation". Further challenges related to the topic of the paper are "Run-time Quality Assurance Techniques" as we provide a framework for collecting monitored execution data; and "Proactive SLA negotiation and agreement" as we contribute to the prediction of QoS parameters before and during negotiation.

Solution / Research Method: The presented solution combines data mining and prediction techniques with semantic technologies and multi-agent systems which together introduce pro-activity and distributed problem solving for increasing the scalability, adaptability and self management of the service provider infrastructure.

Benefits and Evaluation: The solution was tested in a local test environment with 8 hosts. The scheduling and adaptation mechanisms could exploit predictions to increase the availability and response time of the infrastructure.

Relation to Research Framework: The paper may be included in the future in CD-JRA-2.3.7 also, as the presented solution is effective on the infrastructure layer. In that respect, the paper addresses research challenges related to "Supporting adaptation of service-based applications" and "Runtime SLA Violation Prevention". The presented solution provides a generic technique to predict the quality attributes of services and resources, which serves as an enabler to various self-adaptation techniques.

Discussion and Future Work: The work emphasizes on the importance of using historical data by service and cloud providers. The collection and aggregation of monitoring data requires extra implementation effort from the providers. Furthermore, the common format and semantics of historical data collected from various sources is to be investigated in the future.

3.2.4 "Building Dynamic Models of Service Compositions With Simulation of Provision Resources" [25]

Context and Background: Service compositions have been studied thoroughly over recent years and different models to define service compositions have emerged [57]. Approaches like BPEL [55] or YAWL [58] define service compositions in a top down manner using specific notation. At the same time, abstract service composition models include different strands of Petri Nets [16] and process calculi [59].

Key to business usability of service compositions is conformance of their Quality of Service (QoS) attributes with Service-Level Agreements (SLA). Both are intimately related to monitoring and adaptation capabilities [41]. From the computational point of view, resource utilization management, especially the ability to scale computational resources to the expected level of demand, may have drastic impact on response time and failure rates.

Problem Statement: QoS behavior and the SLA constraints that the service provider may be willing to offer to its users depend on two main groups of factors. The first group arises from the structure and logic of the given service composition, and from the QoS attributes of the component services it uses. The second group of factors depend on the capacity and up-/down-scaling policies of the provider's infrastructure on which the provided compositions (and possibly some of the component services execute). To adjust the level of QoS in line with SLA requirements, the provider can try to adapt the composition, the capacity of the provision chain (the infrastructure), or both. While the first approach has been extensively studied, the problem of choosing and triggering adequate policies for adapting the behavior of the provision chain has been largely left open.

This paper tries to address the stated problem by proposing an approach for automatic derivation of dynamic, continuous-time models of behavior of service orchestrations. Those dynamic models are the core for developing simulation models that are used to qualitatively and quantitatively test the behavior of the composition provision chain under various interesting input regimes and to test potential resource management (e.g., up- and down-scaling) policies to assure that the specified QoS levels are met.

Relevance of Problem and Progress from State of the Art: Some aspects of dynamic modeling of service provision chains have already been studied [30]. Such modeling approaches in general apply the tools of system dynamics [54]. We extend the previous work on applying system dynamics to (atomic) service provision management [2], by automatically deriving the quantitative indicators for a service provision chain (number of executing instances, invocation and failure rates) from the structure of a particular composition being provided, as well as from a model of computational resources involved in provision. These quantitative indicators are, in fact, the expected values of QoS attributes for the service composition, derived from the structure of the orchestration, under the given input regime and provision resource model.

We propose an approach that utilizes a common (composition-independent) service provision framework that divides the modeling concern into the composition and the computational resource parts. The former has been studied in several interesting, but specific, cases [30, 46], and some examples of a more systematic approaches to building such dynamic composition models based on QoS constraints have been demonstrated [64]. Our intention is to propose a generic method of converting descriptions of orchestrations in the form of a place-transition networks (PT-nets) [20] into dynamic models, in a manner that ensures composability of orchestrations within choreographies. We believe that such automatic dynamic model generation is a prerequisite for practical application.

Relation to WP Challenges: The paper addresses the challenges “Quality Prediction Techniques to Support Proactive Adaptation” and “Exploiting user and task models for automatic quality contract establishment”.

Solution / Research Method: The starting point is a conceptual model for dynamic representation of service compositions (orchestrations and choreographies). The goal of this model is to represent how the numbers of executing activites and rates of activity-to-activity transitions vary over time. For that, we use a conceptual framework for the continuous-time (CT) composition modeling. The fundamental building block of an *orchestration* CT (OCT) *model* is a *variable* that has a time-varying value. Compared to a PT-net model, *activity variables* are attached to transitions inside, and their value is the expected number of executing instances of a given activity at each point in time. In our example, each service in the data cleaning process would be represented with an activity variable, that shows the expected number of concurrently executing instances of the service at any moment.

The OCT model can be automatically built from the structure, and some behavioral parameters, of a service orchestration. One possibility, described in the paper, is to take a PT-net description of the orchestration, together with branching probabilities, and convert it automatically into a set of ordinary differential equations. Variables that appear in that equation set correspond to places and transitions of the PT-net. We assume an idealized execution environment where tokens, corresponding to process running instances, do not stay for a noticeable period of time in places before triggering a transition. On the other hand, transitions, which correspond to the activities within the orchestration, take some definite time to execute, modeled with an exponential time distribution around the mean.

A dynamic model is good for studying transitional and oscillatory behaviors of the orchestration components under realistic, non-stationary input regimes. Besides, the dynamic model can provide us with insights to the statistical behavior of individual instances and/or sets of instances in different stages of execution, when it comes to the execution time and the expected number of partner service invoked/executing at any moment of time. These can be obtained from the numerically calculated outflows from the OCT model given its in/outflows.

In the model framework we also include a model of computation (service provision) resources. In our example, we use a simple example of service threads, where new machines (hardware or virtual) can be added, and if necessary removed, to scale the computation resources up or down as necessary. The dynamic resource model derives the system load information from the OCT model, and provides the feedback in the form of extra delays and/or blocking. Some aggregate QoS quantities can be directly read from the framework model, given the input regime: the success rate, the failure rate, and the rejection rate (inverse of availability).

Benefits and Evaluation: To evaluate the approach, we have set up the actual running process of medium complexity (including parallel flows, branches and loops), and compared its results with the simulation for median and the average running times of the orchestration. The objective was to establish whether the automatically derived dynamic (ordinary differential equations) model of a service orchestration realistically reflects the execution dynamics of the real process. To that end, we have compared the response of the dynamic model to a Dirac pulse (simulating arrival of a single request) with the empirical distribution of running times, and found reasonably good incidence of the median running times between the two.

Relation to Research Framework: This contribution is related to JRA-1.3 ("End-to-End Quality Provisionand SLA Conformance") and to JRA-2.2 ("Adaptable Coordinated Service Compositions"), of the research framework.

Discussion and Future Work: The approach proposed in this contribution can be used for developing dynamic models of service composition provision, based on an automatically derived continuous-time ordinary differential equation model of the target orchestration. The orchestration model is calibrated using empirical estimates of average activity execution times and branching probabilities, obtained from log analysis, event signaling, or other monitoring tools at the infrastructure level. Several dynamic models of orchestrations provided together can be composed in a modular way.

The resulting dynamic model of composition provision can be used for exploring how the provision system reacts to different input rates (requests per unit of time), testing and choosing different resource management strategies and their parameters, in the style of *management flight simulators*. The model output can be used for both quantitative prediction and qualitative assessment of reference modes (growth, oscillation, stagnation, etc.).

Our future work will concentrate on developing tools that allow simultaneous model calibration/simulation using live monitoring data, along with using these data for orchestration process discovery, when the design and its representation in Petri Net form is not given.

3.2.5 "QoS Assurance for Service-Based Applications Using Discrete-Event Simulation" [26]

Context and Background: The new paradigm for distributed computing over the Internet is that of Web services. The goal of Web services is to achieve universal interoperability between applications by using standardized protocols and languages. One of the key ideas of the Web service paradigm is the ability of building complex and value-added Service-Based Applications (SBAs) by composing preexisting services. For a SBA, in addition to its functional requirements, Quality of service (QoS) requirements are important and deserve a special attention. To assure the desired QoS requirements, different analytical quality assurance techniques can be used. The goal of these techniques is to evaluate QoS and uncover quality defects in the SBAs after they have been created. In this paper,a Discrete-Event Simulation (DES) is adopted to assure QoS of SBAs.

Problem Statement: Being able to characterize SBAs based on QoS has three distinct advantages:

- It allows for the design of SBAs according to QoS requirements. Indeed, it is important for service providers to know the QoS of a SBA at prior before offering it to their clients.
- It allows for the selection and the execution of SBAs based on their QoS. Since many services provide overlapping or identical functionality, different SBAs can be composed, satisfying the same functional requirement. A choice needs to be made to determine which SBA is to be used to provide with the more beneficial QoS.
- It allows for the evaluation of alternative adaptation strategies. The dynamic and unpredictable nature of the execution environment (e.g., network resources and devices characteristics) has an important impact on QoS of SBAs. Thus, in order to better fulfill QoS requirements, it is necessary to adapt SBAs in response to an unexpected evolution of the execution environment.

To assure the desired QoS requirements for an SBA, different analytical quality assurance techniques can be used. The goal of these techniques is to evaluate QoS and uncover quality defects in the SBAs after they have been created. An example for analytical quality assurance techniques is simulation. The goal of the simulation technique is to emulate the conversational behavior of the atomic WSs of an SBA. In this paper, a special case of simulation that is Discrete-Event-Simulation (DES) is used to assure QoS of SBAs.

Relevance of Problem and Progress from State of the Art: To achieve the desired QoS of a SBA, two complementary kinds of techniques can be employed: constructive and analytical quality assurance techniques. The goal of constructive quality assurance techniques is to ensure QoS and prevent the introduction of quality defects while the SBA is created. Examples of such techniques include code generation, software development guidelines, as well as templates. The goal of analytical quality assurance techniques is to evaluate QoS and uncover quality defects in a SBA after it has been created.The analytical quality assurance techniques are sub-divided into three major classes: static analysis, monitoring, and testing. This work deals with using simulation as an analytical testing technique to assure QoS of SBAs.

Relation to WP Challenges: The paper addresses the research challenge of "Quality Prediction Techniques to Support Proactive Adaptation" as it proposes a new analytical testing technique that is Discrete-Event-Simulation (DES) to assure QoS of SBAs. DES allows the prediction of SBA performance in different status and load conditions of the execution environment. The predicted QoS results are used to provide feedbacks on the efficiency of the application.

Solution / Research Method: This work deals with using simulation as an analytical testing technique to assure QoS of SBAs. There are two main reasons for adopting simulation technique: first, simulation is a dynamic analytical technique that allows QoS predictions for software applications in different status and conditions of the execution environment. Second, simulation allows to tune and to evaluate software applications without experiencing the cost of enacting them. The originality of this work is the adoption of a special case of simulation that is the Discrete-Event Simulation (DES). To perform DES, a discrete-event modeling approach for SBAs is proposed. This approach includes a quality model for Web services focusing on essential properties of QoS that play a critical role for the effective management of Web services and a context model that supports an explicit description of the execution environment. This approach is supported by a simulation framework named *SBAS*.

Benefits and Evaluation: To show the effectiveness of our approach, a set of simulation experiments are conducted in order to evaluate and to validate three SBAs that provide the same required functionality.

Relation to Research Framework: The approach focuses on the Service Composition and Coordination layer (JRA-2.2), as service compositions are tested to evaluate QoS properties and to provide appropriate adaptation strategies.

Discussion and Future Work: In future work, the proposed approach can be extended to support dynamic adaptations of SBAs. This requires extensive simulation experiments to define, validate, and enhance different adaptation strategies.

3.2.6 "Enabling Proactive Adaptation through Just-in-time Testing of Conversational Services" [13]

Context and Background: Service-based applications (SBAs) are increasingly composed of third party services which lie beyond the control of service consumers, and can thus change independently or evolve in unanticipated ways. For this reason, third-party services must not only be shown to work during design-time, but should also be (re-)checked to detect potential failures during the operation of the SBA. The detection of a failure should trigger an adaptation to ensure that the SBA maintains its expected functionality and quality.

Detecting failures of third party services is especially challenging when the SBA is making use of conversational (aka. stateful) services. A conversational service is one where meaningful interaction requires more than one operation invocations, and where only specific sequences of operation invocations are accepted (i.e. there exists a protocol).

Problem Statement: The fact that a conversational service might fail in the middle of an invocation sequence can make the adaptation of an SBA quite costly, in both computational and business terms. Substituting a faulty conversational service requires transferring its state to a new replacement service, and possibly initiating some compensation actions for effects which have already taken place. An important research question, therefore, is whether it is possible to avoid such costly operations by being able to determine that such a service is faulty before its time of execution (but as close as possible to that time), such that adaptations can be performed in a proactive manner.

Relevance of Problem and Progress from State of the Art: The paper presents a novel approach for just-in-time testing of the behaviour of conversational services which allows potential failures to be detected shortly before the services need to be executed, and SBAs to be adapted proactively. In contrast to the testing-centric approach described in the paper, most past works relevant to the topic have focused on monitoring as the basic means for identifying failures of an SBA's constituent services. However, monitoring only allows for a reactive approach to adaptation, i.e., the application can be modified only after a failure has been observed. To support proactive adaptation, it is necessary to have some form of prediction of the future functionality and quality of the SBA. The proposed approach differs from past works in the direction of behaviour prediction which have thus far focused primarily on quality attributes rather than protocol conformance, or have dealt with stateless services, rather than conversational ones.

Relation to WP Challenges: The paper contributes to the research goals of WP-JRA-1.3, and specifically to the area of principles, techniques and methodologies for assuring end-to-end quality provision across the chain of service providers and consumers.

Solution / Research Method: The paper advocates performing just-in-time testing of conversational services as a novel approach to detect potential problems in SBAs and to proactively trigger adaptations. The described approach is building on previous work by the authors on a framework for online testing of services (PROSA) and a tool suite for automated generation and execution of service test cases (JSXM) which leverages formal modelling of system behaviour (Stream X-machines).

The paper provides an introduction to the formal underpinnings of the whole approach as well as the available tool support, and details the steps involved for just-in-time testing using a realistic example of an e-shop SBA. An important part of the described process is a new method for contextualising the tests to be carried out on a service, for the specifics of a particular SBA. This involves reducing the number of test cases which are automatically generated from the service's generic model of behaviour, to those relevant for the SBA of interest. The last section of the paper is dedicated to a discussion on the applicability of the proposed approach in terms of underlying assumptions and performance.

Benefits and Evaluation: The presented approach offers a solution to the problem of detecting failures of conversational services after their deployment and before they are executed by an SBA, thus allowing the SBA to be proactively adapted and the failure to be avoided. The work in question differs from the bulk of existing research in this area because it focuses on conversational rather than stateless services, and specifically looking at functional correctness rather than quality of service.

A particular strength of the approach is in the formal method employed for modelling the behaviour of conversational services and generating functional test cases. Compared to other techniques for test case generation, the significant advantage offered by the Stream X-machine testing method is its completeness guarantees.

Another important objective that the paper addresses is ensuring that just-in-time testing can be done with feasible cost and effort, as well as within a reasonable amount of time. To improve the efficiency of online testing the paper proposes a technique to reduce the number of test cases to the minimum which could guarantee the correctness of the service's behaviour in the current SBA/composition context. The paper includes a discussion of this test case reduction method with an experimental evaluation demonstrating time savings during testing which range between 55 and 60%.

Relation to Research Framework: With respect to the S-Cube research framework, the approach presented in the paper is mainly related to the research area of "Quality Definition, Negotiation, and Assurance", and to a lesser extent related to the areas of "Adaptation and Monitoring" and "Engineering and Design".

Discussion and Future Work: Next steps in the context of the S-Cube project include investigating the integration of the presented approach into the PROSA framework (PRO-active Self-Adaptation). Another course of future research could be to investigate the realisation of alternative adaptation strategies in relation to the proposed techniques. For example, if an alternative service is chosen to replace a faulty one, the alternative service may be also (pre-)tested to ensure it provides the same functionality before the actual adaptation will be performed.

3.2.7 "Usage-based Online Testing for Proactive Adaptation of Service-based Applications" [53]

Context and Background: Service-orientation is increasingly adopted as a paradigm for building highly dynamic, distributed and (self-)adaptive software systems, called service-based (or service-oriented) applications (SBAs). An SBA is realized by composing individual software services. In contrast to a software component, for the service composer (aka. service consumer) a software service is not an individual piece of software. Instead, the service consumer can only access the functionality and quality provided by that piece of software via the services interface. There is a clear trend that in the future SBAs will increasingly be composed from third-party services that are accessible over the Internet. SBAs based on third-party services allow taking the concept of ownership to the extreme: not only is the development, quality assurance, and maintenance of the software under the control of third parties, but the software itself is also operated and managed by them. This scenario implies a fundamental change to how software is developed, deployed, and maintained. An SBA cannot be specified and realized completely in advance (i.e, during design-time) due to the incomplete knowledge about the third-party services as well as the systems context and communication infrastructure. Thus, compared to traditional software systems, much more decisions need to be taken during the operation of the SBA (i.e., after it has been deployed), once the missing knowledge is available. One specific problem that needs to be faced in that setting is that third-party services can change or evolve in ways not anticipated by the service consumer. For instance, a service can become unavailable, or can reply too slow due to network latencies or overload at the providers side. This means that SBAs need to dynamically adapt to such failures during run-time to ensure that they maintain their expected functionality and quality. Ideally, the need for an adaptation is proactively identified, i.e., failures are predicted before they can lead to consequences such as costly compensation and roll-back activities.

Problem Statement: Key to proactive adaptation is the ability to predict the future quality of the SBA and its constituent services. Typically, monitoring is used to assess the quality of the SBA and its constituent services during their operation. Based on monitoring data, failures are predicted and thus the need for adaptations is identified. However, monitoring only observes services or SBAs during their actual use in the field. Due to its "observational" or "passive" nature, monitoring does not guarantee a comprehensive coverage of the test object, i.e., monitoring might not cover all relevant service executions. This can diminish the precision of failure prediction, i.e., the ability to correctly predict deviations in expected functionality or quality. To address the shortcomings imposed by the "passive" nature of monitoring, researchers have suggested performing test activities during the operation of the SBA. Such an online testing means that the constituent services of an SBA are systematically tested in parallel to the normal use and operation of the SBA. Thus, "online testing" is sometimes called "active monitoring" in the literature. In online testing, like in traditional testing, we face the problem of determining when, how and how much to test. However, in answering those questions we need to address the following two requirements for online testing, which are imposed by the key differences of SBAs from traditional software systems:

- Services need to be (re-)tested periodically in order to determine failures, because third-party services can change without notice. This need for periodically retesting is significantly different from traditional software systems. A tester of a traditional software system is aware of the changes of software components and only needs to run regression tests after such changes. Furthermore, a tester of a traditional software system has control over the environmental conditions (test environment) during testing and thus, if a test has passed/failed, it will pass/fail again for a later invocation of the same version of the component/system.

- The number of online tests needs to be limited due to economic and technical considerations. This limitation in the number of tests is different from traditional software systems. The number of online tests can affect the provisioning of the service and thus could impact on performance for example. Furthermore, testing costs can become a limiting factor as the software components that provide the service are not owned by the "testing" organization and the service provider might charge "per use" of the service. Finally, the number of times that a user is allowed to invoke a service can be limited by service contracts.

Relevance of Problem and Progress from State of the Art: Recent trends in SOC research show emphasis on proactive adaptation for SBAs, and quality prediction of SBAs has become a vivid research area.

Our approach progresses from the state of the art by devising and evaluating a novel test case selection technique that exploits synergies between monitoring and usage-based testing in order to increase the precision of failure prediction and thus proactive adaptation.

Relation to WP Challenges: The paper addresses the research challenge of "Quality Prediction Techniques to Support Proactive Adaptation" and 'Run-time Quality Assurance Techniques", as the prediction occurs during run-time.

Solution / Research Method: In this paper we address the above problems and introduce a novel online testing technique that provides enhanced proactive adaptation capabilities to SBAs. More specifically, the paper provides the following major contributions:

- A framework for proactive adaptation, which exploits synergies between monitoring, online testing and quality prediction. The frameworks core element is a test selection activity that utilizes information about the usage of the SBAs services in order to select test cases that lead to better coverage of service executions, while utilizing a limited number of online test executions.
- Prototypical implementation of that framework based on an existing monitoring framework with components to collect and report monitoring information, and to execute usage-based online tests.
- Simulation and experimental assessment of the proposed techniques and framework, assessing and analyzing the improvements of our approach on the precision of failure predictions for the case of performance testing, i.e., focusing on response time as a QoS attribute.

Benefits and Evaluation: The complementary use of online testing & monitoring (our approach) improves the precision of failure prediction (i.e., the ability to correctly predict violations in the expected service quality) when compared to monitoring in isolation. Ultimately, this means that an SBA furnished with the complementary approach will have better proactive adaptation capabilities.

To assess the above benefit of our approach, we ran an experiment which is based on (1) the simulation of an example SBA and its associated services, together with (2) the prototypical implementation of the online test case selection, execution and prediction components of our proactive adaptation framework.

Relation to Research Framework: The approach, from a mechanisms point of view, focuses on the Service Composition and Coordination layer (JRA-2.2), as individual services are monitored and tested to determine failures and deviations. In addition, the approach is relevant for Monitoring and Adaptation (JRA-1.2).

Discussion and Future Work: In this paper we addressed quality prediction for proactive adaptation of SBAs. To this end, we presented a framework for proactive adaptation of SBAs based on usage-based online testing. The framework exploits synergies between monitoring, online testing and quality prediction to assure better coverage of service executions, and thus, enables more precise prediction of adaptation triggers. The framework relies on a test case selection component which exploits information about the usage of the SBAs services. This includes techniques to determine the number of test cases to be executed. Furthermore, we introduced our prototypical implementation of the framework with components to collect and analyze monitoring information (including usage frequencies), and to determine and execute usage-based online tests. Finally, we presented the results of an experiment we conducted which showed that the complementary use of testing and monitoring as advocated by our framework, improves the precision of failure prediction (i.e., the ability to correctly predict violations of expected service quality) when compared to using monitoring in isolation.

Several issues remain open for future work. For example, we plan to use more advanced prediction techniques in order to enhance the prediction and to further reduce the number of tests that are needed. We will also consider other different metrics and cost models that would, for example, relate the cost of testing to the cost of compensation activities of wrong adaptations. Furthermore, we plan to run live experiments with executions of SBAs and test cases in order to more realistically assess our approach.

3.2.8 "Towards Proactive Adaptation with Confidence: Augmenting Monitoring with Testing" [38]

Context and Background: Service-based applications (SBAs) need to operate in a highly dynamic world, where services keep changing and evolving. Since SBAs are composed of individual services, SBAs have to react to failures of those constituent services to ensure that they maintain their expected functionality and quality of service. In such a setting, monitoring is typically used to identify failures of the running instance of a service-based application, which in turn triggers the *reactive adaptation* of the failed SBA instance to compensate for those failures.

Monitoring an SBA can only observe changes or deviations after they have occurred, such a reactive adaptation, has several important drawbacks, such as loss of money or increase of execution time (see Section 3.2.11 for more details).

In the case that more than one SBA instance is running, those drawbacks can be avoided. Assume that a failure has been observed in SBA instance A, then other, running SBA instances B and C could be *pro-actively adapted*. This means that instead of adapting B and C after they have failed, those SBA instances could be modified in such a way that they will be prevented from failing due to the same reason that lead to the failure of A.

Problem Statement: When taking pro-active adaptation decisions it is key there is confidence in the prediction of future failures (i.e., certainty that the prediction is correct). This means that pro-active adaptations should not be initiated based on uncertain predictions of failures for A but should be based on statistically sound evidence. Especially, in the case where a service-based application is built from external (third party) services and thus those constituent services are not under the control of the service composer, the observed quality of service and functionality of those constituent services can vary between different service invocations. As an example, a failure observed at one point in time can disappear at a later point in time, as for instance, a service provider could have repaired the service in the meantime. This means that even if a constituent service fails during the execution of SBA instance A, it might well be the case that the service works as expected when invoked for SBA instances B or C. Thus, if uncertain predictions of failures were used as a basis adaptation decisions, this could lead to unnecessary adaptations, such as a re-selection of service provider, for those SBA instances.

Relevance of Problem and Progress from State of the Art: Unnecessary adaptations can lead to severe consequences: Firstly, they can be costly in terms of time and money, even when pro-active actions are taken that are designed to minimize these factors. For instance, additional activities (such as SLA negotiation for alternate services) might have to be performed or the adaptation can lead to a more costly operation of the SBA (e.g., if a seemingly unreliable service is replaced by a more costly one). Secondly, an adaptation could lead to a fault, e.g., if the new service has significant bugs, leading to severe problems as a consequence. Thus, unnecessary adaptations should be avoided as best as possible.

It should be noted that in the case of reactive adaptations, a failed SBA instance is adapted as soon as a failure has been observed. This is also the case for sporadic failures, as in fact, even such a sporadic failure has lead to a problem for the SBA instance. This also means that reactive adaptations can be dealt with existing techniques and thus in this paper we only focus on novel techniques to enable pro-active adaptation.

Currently, there exist several approaches to predict the quality of software and service systems. Many of them consider statistical techniques during monitoring (e.g., log correlation) or during testing (e.g., statistical testing). However, those approaches face a significant shortcoming when applied to adaptive service-based systems. These techniques usually require a high number of data points (monitoring or test results) to produce statistically sound, and confident data, i.e., if not enough data points are available, the confidence for some of the observed failures might not be guaranteed. This poses significant problems for the applicability of those techniques in the SBA context. If monitoring approaches are applied and only few users have started to use the SBA, the collected monitoring data will only sparingly cover the SBA and thus will not suffice to produce confident results. If testing approaches are applied, achieving the required number of data points can lead to significant, additional costs. This is especially true if the SBA includes external services, and thus invoking those services will be associated with additional costs.

Relation to WP Challenges: The paper addresses the research challenge of "Quality Prediction Techniques to Support Proactive Adaptation" and partially "Run-time Quality Assurance Techniques", as the prediction occurs during run-time.

Solution / Research Method: The paper proposes to augment monitoring with online testing in order to produce high confidence failure probabilities. With online testing we mean that the SBA is tested (i.e., fed it with dedicated test data / input) in parallel to its normal use and operation. The online test cases are determined in such a way, that the results provide additional data that complements the data collected through monitoring.

Benefits and Evaluation: The proposed approach is expected to achieve the required number of data points to reach the desired confidence in failure prediction. As we build on the monitoring data available, the testing effort can be kept smaller due to the synergy effects between monitoring and testing. Although the integration of monitoring and testing has been proposed in the literature (e.g., to exploit test cases for monitoring), that novel way of integrating monitoring and testing has not been addressed so far.

The application of our approach has the potential to ensure that all pro-active adaptation decisions can be taken with the desired level of confidence. The failure probabilities can be taken into account during the actual adaptation decisions to determine whether to adapt the SBA or to accept the known risk (i.e., probability) that a failure will occur during the actual operation of the system. In the latter case, the repair of the SBA would still be feasible by following a reactive approach.

The benefits and the applicability of the approach are demonstrated by means of a scenario from the E-Government domain (see WP-IA-2.2).

Relation to Research Framework: The approach, from a mechanisms point of view, focuses on the Service Composition and Coordination layer (JRA-2.2), as individual services are monitored and tested to determine failures and deviations. In addition, the approach is relevant for Monitoring and Adaptation (JRA-1.2).

Discussion and Future Work: The position we took in this paper is that augmenting monitoring data with dedicated test results from online testing promises to produce highly confident failure predictions, based on which pro-active adaptation decisions can be taken with high confidence. In order for this vision of synergetic monitoring and testing to become reality, several important research challenges remain to be addressed, which especially include how to exploit techniques from data mining and statistics to determine the confidence of an existing data set and to compute the additional test cases to reach a desired confidence.

3.2.9 "Towards Proactive Adaptation: A Journey along the S-Cube Service Life-Cycle" [39]

Context and Background: Service-orientation is increasingly adopted as a paradigm for building highly dynamic, distributed and adaptive software systems, called service-oriented (or service-based) systems. This paradigm implies a fundamental change to how software is developed, deployed, and maintained [12]: A service-based system cannot be specified and realized completely in advance (i.e, during design-time) due to the incomplete knowledge about the interacting parties (e.g., third party service providers) as well as the system's context and communication infrastructure [8]. Thus, compared to traditional software engineering, much more decisions need to be taken during the operation of the service-oriented system (i.e., after it has been deployed).

Problem Statement: 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. In order to avoid unnecessary and costly adaptations, the adaptation needs have to be identified more precisely. To enable an automatic adaptation, the relevant artefacts, 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 artefacts the approach identifies the need for an adaptation proactively.

Relevance of Problem and Progress from State of the Art: So far, the major work on adaptation has been centered around reactive adaptation capabilities based on monitoring [5]. This means that adaptation is performed *after* a deviation or critical change has occurred. Such a reactive adaptation based on monitoring, however, has at least the following two important shortcomings (cf. [38, 22] and [17]):

- It can take time before problems in a service-based system lead to monitoring events that ultimately trigger the required adaptation. One key trigger for an adaptation should be the case when the service-based system deviates from its requirements (such as expected response time for example). If only those requirements are monitored (e.g., see [3]), the monitoring events might arrive so late that an adaptation of the Service Based Application (SBA) is not possible anymore. For instance, the system could have already terminated in an inconsistent state, or the system has already taken more time than required by the expected response time.
- Reactive adaptation can become very costly, especially when compensation or rollback actions need to be performed. As an example, when using stateful (aka. conversational) services [14], the state of the failed service might need to be transferred to an alternative service.

Of course, one can monitor the individual services of an SBA and trigger an adaptation as soon as the service has failed, i.e., violated its contract [19]. However, when using those techniques it remains unclear whether the failure of this service could lead to a violation of the SBA's requirements. This means that there may be situations in which the SBA is adapted although it would not have been necessary, because the requirements might still have been met. Consider the following simple example: Although a service might have shown a slower response time as (contractually) expected, prior service invocations (along the workflow) might have been fast enough to compensate the slower response of that service.

Such unnecessary (or "false positive") adaptations have the following shortcomings [38]:

- Unnecessary adaptations can lead to additional costs and effort that could be avoided. For instance, additional activities such as Service Level Agreement (SLA) negotiation for the alternative services might have to be performed, or the adaptation can lead to a more costly operation of the SBA, e.g., if a seemingly unreliable but cheap service is replaced by a more costly one.
- Unnecessary adaptations could be faulty (e.g., if the new service has bugs), consequently leading to severe problems.

In summary, one key problem that needs to be solved to enable proactive adaptation is to determine whether the service-based application, during its future operation, might deviate from its requirements.

Relation to WP Challenges: The paper addresses the research challenge of "Proactive SLA negotiation and agreement", "Run-time Quality Assurance Techniques" and "Quality Prediction Techniques to Support Proactive Adaptation".

Solution / Research Method: Adaptive SBAs automatically and dynamically adapt to changing conditions and changes of service functionality and quality. To enable such 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. In this approach a formalization is introduced, as well as techniques that build on this formalization.

Benefits and Evaluation: For evaluating the process model proposed by the approach, we specify the required artefacts. 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, the identify the need for identification. We run the validation on a Windows 7/x86 platform. 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. 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.

Relation to Research Framework: The approach focuses on the Service Composition and Coordination layer (JRA-2.2). Individual services are monitored in order to determine failures and deviations. In addition, the approach is relevant for Monitoring and Adaptation (JRA-1.2), as it identifies adaptation needs and hence triggers adaptations.

Discussion and Future Work: We are currently striving to push the envelope towards proactive adaptation even further. In addition to determining the need for adapting the service-based application based on actual failures of the application's constituent services, we investigate the applicability of online testing for predicting the quality of those services (e.g., see [38, 22]). Combined with the introduced approaches, this means that critical problems could be observed even earlier, thus enabling a broader range of adaptation and evolution strategies. For instance, in our running example we can only react to the violation of the response time of a constituent service by ensuring that the remainder of the workflow executes faster. However, if the quality prediction techniques forecast a violation of the expected response time of a specific service, this very service can be replaced before it is invoked in the context of the service-based application.

3.2.10 "A Soft Constraint-Based Approach to QoS-Aware Service Selection" [63]

Context and Background: Establishing QoS contracts, described in the Service Level Agreement (SLA), that can be monitored at runtime, is therefore of paramount importance. Various techniques [43] to select services fulfilling functional and non-functional requirements have been explored, some of them based on expressing these requirements as a constraint solving problem [44, 11] (CSP). Traditional CSPs can either be fully solved (when all requirements are satisfied) or not solved at all (some requirements cannot be satisfied). In real-life cases, however, over-constraining is common (e.g., because available services offer a quality below that required by the composition), and problems are likely not to have a classical, crisp solution. The aim is to try to relax and predict quality of the constraints.

Problem Statement: Service-based systems should be able to dynamically seek replacements for faulty or underperforming services, thus performing self-healing. It may however be the case that available services do not match all requirements, leading the system to grind to a halt. In similar situations it would be better to choose alternative candidates which, while not fulfilling all the constraints, allow the system to proceed.

Relevance of Problem and Progress from State of the Art: Traditional CSPs can either be fully solved (when all requirements are satisfied) or not solved at all (some requirements cannot be satisfied). In real-life cases, however, over-constraining is common (e.g., because available services offer a quality below that required by the composition), and problems are likely not to have a classical, crisp solution. Solving techniques for soft CSPs (SCSP) can generate solutions for overconstrained problems by allowing some constraints to remain unsatisfied.

Relation to WP Challenges: The work presented in the paper is related to the quality prediction challenge. In fact, the proposed work predicts the quality by relaxing the constraints. We apply soft constraints to modelling SLAs (including penalties for unfulfilment of duties) and to decide how to rebuild a composition even in the case that not all the requirements can be satisfied. The system is therefore able to self-heal in such a way that it gracefully degrades as service quality does instead of stopping suddenly.

Solution / Research Method: We study the use of soft constraints in this scenario. Soft constraint theory allows constraints to be ranked and not to be wholly satisfied, while giving a solution as optimal as possible. *Soft constraints*, instead of the traditional *crisp* constraints, can help naturally model and solve replacement problems of this sort. In this work we apply soft constraints to model SLAs and to decide how to rebuild compositions which may not satisfy all the requirements, in order not to completely stop running systems.

Moreover, our framework takes into consideration the penalties agreed upon on the SLA by building a new (Soft) Service Level Agreement (SSLA) based on preferences where strict customer requirements are replaced by soft requirements allowing a suitable composition. This agreement has to include penalty terms to be applied while the contract terms are violated.

Benefits and Evaluation: The benefits of the proposed solution is the fact the approach does not only provide the optimal solution, but all possible solutions. Hence, we have to retrieve all possible solutions with their preference values. Then,we have to rank those solutions according to their preference values. What is to be noticed is that we have replaced the constraint concept by the preference concept. Both of the two concepts are closely related notions, since preferences can be seen as a kind of tolerant constraint. While using Soft Constraint Satisfaction Problem, a preference value for choices weight could have one or more elements. A possible solution is to rank the elements according to other criteria. In addition to that, the system has to take into consideration some other information such as penalties. Up to now any work is discussing penalties in such a way. We build a new Service Level Agreement (SSLA) based on preferences. A prototype of the framework is ongoing.

Relation to Research Framework: The work is related to JRA 1.2 monitoring while the quality is monitored in order to select service for the composition JRA 2.2.

Discussion and Future Work: We have presented a soft constraint-based framework to seamlessly express QoS properties reflecting both customer preferences and penalties applied to unfitting situations. The application of soft constraints makes it possible to work around overconstrained problems and offer a feasible solution. Our approach makes easier this activity thanks to ranked choices. Introducing the concept of penalty in the Classical SCSP can also be useful during the finding and matching process. We plan to extend this framework to also deal with behavioral penalties.

3.2.11 "Towards data-aware qos-driven adaptation for service orchestrations" [24]

Context and Background: Adaptation in SBAs happens at the moment in which a (serious) deviation w.r.t. the expected behavior is detected. A common type of adaptation is to select an alternative service provider which, while giving semantics compatible with the one required, behaves better in other respects (cost, speed, etc.). We tackle the problem of identifying from a set of available services, the most promising services are identified on the basis of the availability of *functions* which describe their expected behavior based on the characteristics of input data (the request that initiates their execution).

Problem Statement: We assume that services (and service compositions) have an observable behavior which depends on the input data: size, composition, etc. The complexity of such behavior (time, number of partner invocations, etc.) is represented with functions over (abstractions of) data in requests. We want to be able to decide dynamically the best partner for a given request (or set of requests) using these functions, and we also want to automatically synthesize and update the description of a composition based on its structure and on the functional description of the partners it interacts with.

Relevance of Problem and Progress from State of the Art: A more informed adaptation can produce compositions which are fitter in the sense that they incur smaller computational cost and communication overhead, and which therefore are able to give a better final QoS. In our case, this extra information comes from taking data characteristics into account and making an offline analysis which synthesizes abstract descriptions of compositions based on the input data, the composition structure, and the descriptions of the partner services. Tackling data as a first-level citizen has long been neglected in the field of SOC, and there are cases where taking it into account would be clearly advantageous.

Relation to WP Challenges: This work is related to:

- *Run-time Quality Assurance Techniques:* A combination of offline processing and run-time decision procedures are used to provide a better QoS. This is clearly shown in the simulations reported in the paper.
- *Quality Prediction Techniques to Support Proactive Adaptation:* although this paper does not address directly proactive adaptation, the functions and abstractions of behavior can be used (as shown in a companion paper) to predict misfits between the expected and the actual execution ahead of time. Also, the finer grain knowledge about the behavior of services can help in making a better decision which may lower the possibility of future mishaps.

Solution / Research Method: Data-aware static analysis is applied to orchestrations, whose abstraction is automatically generated. These abstract descriptions are used in a simulation where an agent looks for partners to meet a minimum QoS. The proposed method is compared with a random selection and a selection based on a static ranking based on average past behavior of services profiles.

Benefits and Evaluation: The simulations have shown large advantages of the data-aware adaptation approach using abstract behavior descriptions, when compared with any of the other two selection methods (fixed and random selection). Therefore, we can evaluate the proposed approach very positively.

Relation to Research Framework: The approach focuses on making predictions of characteristics which have a direct effect on quality. This is also relevant for monitoring and, as shown in this paper, for smarter adaptation, as a means to obtain better overall QoS. Therefore it is relevant for JRA-1.2. As the underlying techniques can also be used to make a more informed decision between different composition possibilities, it can also contribute to JRA-2.2.

Discussion and Future Work: A salient aspect to be worked on is to have more accuracy in the static analysis of data as commonly presented in SBAs (i.e., in XML format). Specific abstraction techniques would have to be devised for this. Tackling in full the complexity of existing orchestration languages (e.g., BPEL or other) remains as future work.

3.2.12 "Test Coverage of Data-Centric Dynamic Compositions in Service-Based Systems" [23]

Context and Background: One of the defining characteristics of services is their composability, i.e., the possibility to combine individual services to create a new functionality. The de-facto standard for creating Web service compositions is the Business Process Execution Language for Web Services (WS-BPEL). The mechanism of dynamic binding in SBSs allows to define a required service interface at design time and to select a concrete service endpoint from a set of candidate services at runtime.

Service-based systems require thorough testing, not only of the single participants but particularly of the services in their interplay. Initial testing of a dynamic composite service plays a key role for its reliability and performance at runtime. For one thing, the tests may reveal that a concrete service cannot be integrated at all because the results it yields are incompatible with any other service in the composition. For another thing, the test outcome can assist in the service selection process, favor certain well-performing service combinations and avoid configurations for which the tests failed. Furthermore, upfront testing makes it safer and possibly faster to move to a new binding when a new service implementation becomes available in the running system.

Problem Statement: This paper addresses the problem of integration testing of data-centric dynamic compositions in service-based systems. These compositions define abstract services, which are replaced by invocations to concrete candidate services at runtime. Testing all possible runtime instances of a composition is often unfeasible. We regard data dependencies between services as potential points of failure, and introduce the k-node data flow test coverage metric. Limiting the level of desired coverage helps to significantly reduce the search space of service combinations to test.

Relevance of Problem and Progress from State of the Art: Recent trends in service computing research and practice show an emphasis on service composition, and testing of service compositions has become a vivid research area. We are focusing on the particular aspect of data dependencies between services, as these dependencies have a major influence on the overall composition and constitute a potential point of failure. This work contributes to the state of the art by defining test coverage goals based on a formalization of the service composition model, and by presenting an end-to-end solution for testing dynamic service compositions in WS-BPEL.

Relation to WP Challenges: The paper addresses the research challenge of "Run-time Quality Assurance Techniques" and partially "Proactive Adaptation and Predictive Monitoring", because necessary adaptation actions are derived from functional problems determined in the testing procedure.

Solution / Research Method: The paper uses an abstracted view of compositions that takes into account the data flow occurring between individual invocations in a service composition. An exact model of data-centric compositions is presented, based on which the k-node data flow coverage criterion is formalized. We formulate the problem of generating a minimum set of test cases as a combinatorial optimization problem. The service composition model is then mapped to the data model of FoCuS, a coverage analysis tool developed at IBM. FoCuS can efficiently compute near-optimal solutions, which are then used to automatically generate and execute test instances of the composition. The implementation is integrated into TeCoS, a framework that stores service metadata and logs invocation traces in order to generate and execute composition test cases.

Benefits and Evaluation: The introduced criterion of k-node data flow coverage in service compositions greatly reduces the number of test cases and provides for feasible testing of data-centric compositions with dynamic binding and exchangeable service endpoints. The approach has been successfully evaluated in an end-to-end setting with several scenario WS-BPEL service compositions in different sizes.

Relation to Research Framework: The presented approach focuses on the Service Composition and Coordination layer (JRA-2.2), because it focuses on testing and validating data-centric dynamic service compositions. In addition, the approach is relevant for End-to-End Quality Provision and SLA Conformance (JRA-1.3).

Discussion and Future Work: As part of the ongoing work, the framework will be extended to support alternative service composition techniques, particularly focusing on the emerging field of data-centric service mashups. We also strive to further improve the test generation algorithm, and to let the tester control the characteristics (e.g., maximum execution time) of the optimization. Furthermore, the plan is to parallelize the test case execution on multiple server instances.

3.2.13 "A Framework for Proactive SLA Negotiation, Proactive SLA Negotiation for Service Based Systems" [36, 37]

Context and Background: A service level agreement (SLA) is an explicit formal contract between the provider and the consumers of a service that sets the quality and sometimes (functional) properties that should be guaranteed during the provision of the service and the penalties in case of defaulting. SLA negotiation is the process through which service providers and consumers reach an agreement about an SLA. As in any complex application, most services are composed hierarchically, i.e. a service provider needs to access one or more services to offer a specific service. In such settings several SLAs need to be agreed by the participating parties contributing to the final service delivered to the requester. Inability of any of the participating party to meet the service level objective of the agreed SLA may affect the overall service, e.g. suspension of the service provisioning or premature termination of the agreement. There are a number of possible scenarios that may lead to the violation of a service level agreement. More specifically a SLA may be violated because, i) poor quality of service delivered by a participating service, ii) delayed delivery of service by a participating service, iii) unavailable service or resource, iv) change of requester's circumstances, e.g. requester needs service at better level than the agreed level after the SLA has been agreed, v) change of provider's circumstances, e.g. provider suffers from peak demand of services from its requesters. These scenarios suggest that once an SLA is agreed by all the parties and enforced, it becomes necessary to monitor the quality of the delivered service against the service level objectives specified in the SLA and to handle any violation in the SLA detected by the monitor.

Problem Statement: In order to achieve the runtime operation of service based applications with minimised interruptions it is necessary to proactively handle the violation of agreed SLAs, i.e. agreed SLAs should be renegotiated or a new SLA should be negotiated by the participating parties prior to a foreseen problem. From service requester's perspective, proactive SLA negotiation should be able to cover the following scenarios: (a) Following forecasts for a higher level of requester demand that can put its agreed guarantee terms at risk. (b) If the service requester identifies a change of reputation of the service provider. For example, let the SLA established between service provider (SP) and the service requester (SR) and SP is providing the service to SR at the agreed level. Meanwhile SP's overall reputation (say availability) falls due to failure to another service requester SR1. In such case SR may consider to start renegotiation. (c) Service requester may request for renegotiation if it becomes known that competitors provide are going to provide better offerings. (d) Violation of soft constraints in the guarantee terms may trigger the requester to renegotiate.

On the other hand, from service provider's perspective, proactive SLA negotiation should be able to cover the following scenarios: (a) Following forecasts for a higher level of requester demand that can put its agreed guarantee terms at risk. (b) Following forecasts that its constituent services will fail or violate their own guarantee terms (this applies to both software and infrastructure services). (c) Following the forecasts of unavailability of uncontrollable resources (e.g. network bandwidth). (d) Due to the change of other SLA of service provider. Say SP has two requesters SR1 and SR2 where SR1 enjoys higher privilege than SR2. If the SR1 request for more resources and SP decides to provide it to SR1 (since SR1 has higher privilege than SR2), then SP may need to renegotiate SLA with SR2 (resource utilization).

Relevance of Problem and Progress from State of the Art: Several efforts are found in the literature to handle the violation of an agreed SLA. For example, provision of penalties is argued depending on the importance of the service level objectives. Runtime renegotiation is suggested, where either the service level objectives of the agreement are revised to accept service from the existing provider or a new SLA is provisioned with a new service provider terminating the existing SLA. All these approaches are reactive in nature that offers corrective actions only after a service level agreement has been violated. These either affect the quality of the delivered service or fail to guarantee uninterrupted service. In this paper we propose a framework that for proactive SLA negotiation that integrates this process with dynamic service discovery and, hence, can provide integrated runtime support for both these key activities which are necessary in order to achieve the runtime operation of service based applications with minimised interruptions. More specifically we argue that an alternative service provider should be selected and an SLA should be negotiated by the participating parties prior to a foreseen problem and when the existing SLA is violated the faulty service provider should be replaced by the newly selected service provider.

Relation to WP Challenges: This paper addresses the research challenge “Proactive SLA negotiation and agreement” and also partially addresses the challenges “Adaptation and Monitoring” and “Discovery and Registry Infrastructure” as we combined service discovery and monitoring to facilitate the proactive SLA negotiation.

Solution / Research Method: This paper proposes a framework that integrates service discovery and monitoring of service in order to facilitate proactive SLA negotiation. The service discovery process is used by service consumer applications in order to identify potential alternative services for the services that they currently use. The identification of alternative services is based on various characteristics of the published service such as structural, behavioural and QoS specified by the service consumer. The negotiation process is carried out according to a two-phase protocol that may result in a provisionally agreed SLA but not activated SLA or negotiation failure. A provisional SLA is a service level agreement that has been agreed by service consumers but has not been activated yet. Such SLAs may be open ended or have an expiry date by which they will either have been activated or cease to exist. In the negotiation process, desired level of service and other individual or collective issues (e.g. penalty, SLA time frame) are negotiated with each alternative service. It should be noted that a service consumer may specify the bottom line of various QoS characteristics in the service discovery query that enables the service discovery process to identify potential alternative services. But in the negotiation phase, each alternative service is negotiated over the QoS characteristics in order to identify a service that fulfils the service consumer’s (and also service providers’) requirements in optimal way. The monitoring process monitors the runtime behaviour of the service provider and the service requester in order to detect if the agreed SLA is satisfied. If the monitor detects violations of the SLA or detect situations that requires proactive negotiation then the monitor triggers the framework to initiate a renegotiation.

Benefits and Evaluation: The objective of proactive SLA negotiation is to ensure that a service, which could be potentially used by a service consumer, will have an agreed set of guaranteed provision terms if the need to deploy it arises at runtime. Hence, when this need arises it won’t be necessary to engage in a lengthy negotiation process interrupting the operation of the service consumer application.

Relation to Research Framework: The approach presented in this paper focuses on the proactive SLA negotiation (JRA-1.3). Moreover, this approach performs the monitoring of agreed SLAs to enable proactive negotiation that facilitates the proactive adaptation of a service based system (JRA-1.2). In this regard this approach also contributes to the integrated adaptation scenarios presented in deliverable JRA-1.2.5.

Discussion and Future Work: In this paper we present a framework that combines service discovery and service monitoring to support proactive SLA negotiation among participating parties. However the presented framework has also opened broad scope of future investigations. For example the framework can be extended to support proactive negotiation for hierarchical SLA i.e. a complex SLA can be decomposed into several SLAs and negotiated separately to come to a final agreement. Also in the presented framework, negotiation rules are specified by the participating parties before the negotiation starts and followed in the negotiation process. The framework can be extended to support dynamic adaptation of the negotiation rules, i.e. the participants will be able to dynamically change the negotiation rules during the negotiation process.

3.2.14 "A Survey on Service Quality Description" [28]

Context and Background: Quality of service (QoS) can be a critical element for achieving the business goals of a service provider, for the acceptance of a service by the user, or for guaranteeing service characteristics in a composition of services, where a service is defined as either a software or a software-support (i.e., infrastructural) service which is available on any type of network or electronic channel. Nowadays, the literature includes lots of approaches to describe QoS at different levels and they are often independent each other. Thus, a shared vision about the QoS topic is not available.

Problem Statement: The goal of this paper is to compare the approaches to QoS description nowadays presented in the literature, where several models and meta-models are included. The survey is performed by inspecting the characteristics of the available approaches, to reveal which are the consolidated ones and to discuss which are the ones specific to given aspects, and to analyze where the need for further research and investigation is. The approaches here illustrated have been selected based on a systematic review of conference proceedings and journals spanning various research areas in Computer Science and Engineering including: Distributed, Information, and Telecommunication Systems, Networks and Security, and Service-Oriented and Grid Computing.

Relevance of Problem and Progress from State of the Art: The term 'quality document' is used in a generic way as a description of quality, and all issues related to managing such a document in a specific system architecture are not addressed. Even not referring to specific architectural solutions, the paper shows that the problem of being able to write quality documents is far from solved. In the literature, several approaches have been proposed, and there is no commonly agreed way to specify QoS. This paper aims to systematically and comparatively review these approaches according to various criteria, which include their scope, formality, expressiveness, and applicability. Many important findings are uncovered from the analysis. In addition, areas for further research and investigation are spotted.

Relation to WP Challenges: The survey addresses the challenge of "End-to-End Quality Reference Model".

Solution / Research Method: The presented approaches have been selected based on a systematic review of conference proceedings and journals spanning various research areas in Computer Science and Engineering including: Distributed, Information, and Telecommunication Systems, Networks and Security, and Service-Oriented and Grid Computing.

Benefits and Evaluation: The survey aims at providing a systematic literature review of the current approach for describing the QoS for Service Based Applications. The work considers this issues during the entire service life-cycle: starting from the Service Advertisement, to the possible Adaptation.

Relation to Research Framework: This work is mainly related to the Quality of Service Definition (JRA-1.3).

Discussion and Future Work: This paper has focused on investigating the issue of service quality description. To this end, a systematic review of a large number of approaches has been conducted in order to reveal their strengths and weaknesses and highlight where the need for further research and investigation is.

3.2.15 "A dynamic privacy model for web services" [40]

Context and Background: Nowadays, Web services are being recognized as an emerging platforms to quickly develop complex distributed applications. Many services (e.g., mortgage approval, travel agency) require service requesters to disclose some personal data (e.g., credit card number, home address). As the number of inappropriate usage and leakage of personal data is increasing, privacy concerns is becoming one of the most important concerns of service requesters, service providers and legislators to ensure a QoS.

Problem Statement: In order to take into account the privacy concerns of the individuals, organizations (e.g. Web services) provide privacy policies as promises describing how they will handle personal data of the individual. However, privacy policies do not convince potential individuals to disclose their personal data, do not guarantee the protection of personal information, and do not provide how to tackle the evolution of the policies. In this paper, we introduce a framework based on an agreement as a solution to these problems.

Relevance of Problem and Progress from State of the Art: Today, the individuals are becoming more and more concerned about the privacy of their personal data. These concerns might lead to a situation where the customers do not trust the web service any more and take their business somewhere else. So, the important enabling factor for a well usage of online services is building customers confidence with service providers when the latter comes to handle their personal data. Privacy policies are used by web services in order to ease the privacy concerns of their clients and to adhere to legislative measures, stating what they would do or not with the personal information of their clients. The most significant effort currently underway to enable web site users to gain control over their private information, is the Platform for Privacy Preferences (P3P) , developed by the World Wide Web Consortium (W3C). Furthermore, the privacy policies defined in P3P are only promises and do not guarantee the protection of personal information of data subject. There is a need for something more legal than promises a contract-based privacy.

Relation to WP Challenges: The work presented in the paper is related to the "proactive SLA negotiation and agreement"research challenge. A non functional QoS negotiation is handled take into account the evolution of the SBA environment.

Solution / Research Method: We propose a privacy agreement model that spells out a set of requirements related to consumers privacy rights in terms of how service provider must handle privacy information. We define two levels in the agreement (1) policy level (2) negotiation level. A formal privacy model is described in the policy level. We provide upon it a reasoning mechanism for the evolution.

The framework supports in the negotiation level of the agreement a lifecycle management which is an important deal of a dynamic environment that characterizes Web services. Hence, the privacy evolution is handled in this level. A negotiation protocol is proposed to enable ongoing privacy negotiation to be translated into a new privacy agreement.

Benefits and Evaluation: In order to support the privacy agreement negotiation model described in this paper, we devise an architecture of a system providing an explicit management of the privacy agreement. In fact, in the architecture of the privacy agreement, the management system is implemented by the negotiation model between signing parties to manage the behavior of services when possible events may happen and also their interactions. Developers can visually edit privacy agreement model and generate a negotiation. We focused on providing tools to support the negotiation as well as the detection and analysis of relevant events in the dynamic environment of web services. Evaluations and performances have been provided to validate the prototype.

Relation to Research Framework: The work is related to JRA 1.2 monitoring WP, since the privacy agreement is monitored in order to check the privacy agreement compliance. The work is tackling the policy evolution and a dynamic negotiation.

Discussion and Future Work: The main contribution of the work presented in this paper is a framework that leverages the privacy and contract representation technologies and established modeling notation (state machine-based formalism) to provide a high level support for describing the private data use flow as well as the evolution of privacy in Web services. We proposed a formal model for privacy called privacy agreement which is an extension of WS-Agreement specifications, that both service customer and provider might agree before any process is running. We argue that endowing privacy in service with abstractions have benefits and is useful in several situations to automate a non functional activity in the web service. We have emphasized a lifecycle of privacy which is an important issue to date which has not been addressed. The proposed framework with such abstractions shows the automation process of the privacy lifecycle activities that can be greatly enhanced. Future work around our framework will be considering the extension of the proposed approach by the refining and introducing a reasoning mechanism for the temporal aspect about agreement that may change over the time in the agreement negotiation protocol. We will also investigate the expansion of the framework by handling the composition of the services in the privacy agreement.

3.2.16 "Enhancing Service Network Analysis and Service Selection using Requirements-based Service Discovery" [62]

Context and Background: Web services have been serving as an intermediary platform to many organizations which adopted SOA principles with the aim of exposing their applications over the web [61]. This gives rise to service ecosystems which are electronic marketplaces and collection of Web services whose exposure and access are subject to constraints [4]. Service Networks (SNs) are considered as a prominent example of service ecosystems and they have been proposed to model, analyze and optimize business collaborations among various participants by modeling their interactions using networks [9]. SNs reside on a high abstraction business level depicting partners as nodes and their offerings as edges, while hiding details regarding the concrete interactions in terms of business processes. Until now, SN techniques mainly deal with analysis of networks formed out of long-term relationships and trust relationships between partners in order to measure the overall quality of the network and the possible advantages of a member participating in the network, excluding aspects like quality properties of services involved.

Problem Statement: Current SNs are static meaning that all economic entities participating in the network have established long-term relationships with each other. In wider spanning service ecosystems, several service providers may offer functionally replaceable services that differ in their quality characteristics. Service ecosystems should therefore explicitly support the negotiation process, reducing non-critical human involvement and providing decision-makers with the information they require to formulate and assess service offers. In particular, automated support for negotiation over services is needed for comparing requirements of prospective service consumers against capabilities and qualities of available services. This calls for techniques and tools supporting the detection of potential service guarantee violations that will ultimately result in an improved service selection process. This paper addresses the problem of selecting for the consumer the most valuable online service, according to the consumer preferences combining techniques from SN analysis and selection of services based on QoS.

Relevance of Problem and Progress from State of the Art: Someone outside the ecosystem would observe an almost chaotic, random and continuous creation and destruction of service provider and service consumer links. As time progresses, some links occur more frequently than others, indicating a preference of the consumers for some of the services offered. SNs are formed when these links endure over time and they may be simple networks meaning that there is a single link connecting one service provider with one service consumer, or they may be complex enough in a sense that the service provider composes services from several other service providers to offer his own service. Previous works on the service networks field [7] [6] deal with static SNs in which links indicate long-term partnerships and relationships. Analysis techniques on SNs mainly provide quantitative models of value computation for value exchanges in SNs [9] supporting decisions in a strategic level (e.g., how to restructure network, whether to leave network and join other and so on). This work deals with emergent SNs where no stable long-term links are established yet, so the service consumer has no control over the service provision and implementation and to this extend, assessing QoS is vital in establishing trust in a service. It is therefore desirable for the service consumer to be able to clearly state QoS requirements and identify the level of service compliance with those requirements. The research in this paper draws on previous research in requirements-based and ontology-driven service discovery as well as on established SN analysis techniques.

Relation to WP Challenges: This work is partially related to "Run-time Quality Assurance Techniques" and "End-to-End Quality Reference Model".

Solution / Research Method: The key contribution of this paper is a process model for KPI-driven service discovery to enhance SN analysis and service selection. This paper proposes a process to (1) discover candidate services based on quality of service characteristics (QoS) and use them to dynamically identify various classes of service providers to create new Service Networks (SNs) within a service ecosystem, and (2) exploit newly created SNs to improve the service discovery and selection process. With the requirements-based service discovery tools and techniques it is possible to get an early indication of the quality of available services by matching non-functional requirements to service qualities during service selection. In respect to the above goals, Key Performance Objectives (KPOs) model is constructed, capturing business goals of the network's participants, and the service consumer's requirements. This KPO model is then mapped to requirements types based on the predefined rules, which in turn are used to find QoS metrics for candidate services. Monitoring is taken place to measure KPIs for the selected services in order to improve service discovery and selection by filtering out those services (whose specified QoS does not match the measured KPIs) that do not comply with the KPO model and replacing such services with newly published candidate services that comply with the KPO model. In order to increase the overall customer satisfaction, we monitor how the changes influence the service consumer's opinion on each consumed service by calculating the overall customer satisfaction using ACSI. This paper is limited to describe the approach of the usage of requirements-based discovery in SN analysis and service ecosystem modeling and contributes to the research on how to achieve viable and adaptive, dynamic evolving service ecosystems.

Benefits and Evaluation: This paper is an initial step towards analyzing emergent service networks and combines consumer-related information, such as customer demands and feedback, with methodologies regarding requirements-based discovery and selection of services based on their quality of characteristics. The proposed methodology enhance SN analysis by matching KPIs to service qualities for identifying suitable services while taking into account the customer preferences. The American Customer Satisfaction Index (ACSI) [1] has been applied to the discovery process enabling monitoring and selection of services effectively. This will eventually enable a more competitive market because an increase or decrease of customer satisfaction will encourage service providers to react according to user feedback. The evaluation of the idea proposed in the paper has been done using an example of a car manufacturer and its service system in order to show how service delivery monitoring and service brokers can be used to improve consumer satisfaction.

Relation to Research Framework: The approach presented in this work covers some aspects with respect to research framework such as “Business Process Management”, “Quality Definition, Negotiation and Assurance”, and ‘Engineering and Design”.

Discussion and Future Work: This paper reports on the early results from ongoing work into the use of requirements-based service discovery in SN analysis and service ecosystem modeling and it is limited to describing the approach and demonstrating it using an example. Next stages will include evaluation of the process model’s utility and to complete the implementation of the first version of the KPI-driven service discovery engine. We will study the impact of user models on the emergence of new SNs and on the enhancement of the service ecosystem as user modeling can support personalization of services for adaptation to the user. The proposed process will be extended to consider user models to ensure that different classes of users will be able to select candidate services and service providers that facilitate their personal characteristics or preferences. The exact ways that user profile and models will affect service discovery and selection as well as their impact on SNs is work in progress and we will present our results in the near future.

3.2.17 “A semantic based framework for supporting negotiation in Service Oriented Architectures” [10]

Context and Background: In Service Oriented Architectures, the matchmaking activity can be followed by a negotiation activity that allows to identify the Service Level Agreement (SLA) as a composition of Service Level Objectives (SLOs) (i.e service quality bounds) of the delivered service, the penalties enforced when these are violated, the measurement and evaluation process of the SLOs, the price to be payed, and other relevant terms. Focusing on the negotiation, current solutions start from the assumption that all the negotiating parties agree and have the capabilities to support a prespecified negotiation protocol. Although this assumption is mandatory in a negotiation system, in SOA service requesters and service providers might not know each other in advance. This means that it is quite probable that negotiation cannot be enacted, since a common supported negotiation protocol is not easy to be found.

Problem Statement: This work aims at proposing a semantic-based framework for supporting the negotiation in SOA where the involved parties could not be able to perform all the activities required by the negotiation protocol. The framework assumes that all the parties specify their requirements and offers according to the same model, and a matchmaking algorithm to compare them exists. The proposed framework completes the picture by proposing a semantic-based model for allowing the participants to express their negotiation capabilities. When a negotiation is required, the framework can be used to identify which are the possible negotiation protocols by comparing the negotiation capabilities exposed by the participants.

Relevance of Problem and Progress from State of the Art: The need for automated negotiation of quality SLAs and contracts, is being addressed as one of the main driver for the adoption of service based systems in real-world scenarios. However, the literature has usually focused only on the negotiation enactment phase. In this context, the paper identifies two different approaches for the enactment of Web service QoS negotiation, i.e., the broker-based and the agent-based approaches. With the broker-based approach, the execution of negotiation is delegated by service providers and customers to a trusted broker. Conversely, in the agent-based approach each negotiation actor defines its own agent, which embeds the actor's negotiation strategies. In both cases, the structure of the negotiation protocol is usually known a priori by the participants. Moreover, the ability of the broker to execute a protocol or of negotiation agents to enact a given negotiation protocol is taken for granted.

Relation to WP Challenges: The paper addresses the research challenges of "Proactive SLA negotiation and agreement" and "Exploiting user and task models for automatic quality contract establishment".

Solution / Research Method: The negotiation capabilities in the proposed negotiation framework are defined in terms of the ontology that extends OWL-Q, a semantic quality-based service description language. In this work, the ontology is limited to the concepts required for the discovery of a Negotiation Protocol compatible with the negotiation capabilities defined by the potential participants. The focus of the paper is therefore on capturing in an ontology actors, roles, and possible actions that an actor can perform or that a role may require in a given negotiation protocol.

Benefits and Evaluation: In addition to the usual advantages of the use of an ontology, another advantage is that term matching rules can be created in cases where domain experts differently interpret the ontologys terms. Using this framework service requesters and providers are able to describe which are the negotiation protocol they are able to carry out and what are the actions they can perform. In case a negotiation is required, by an ontology reasoning, the framework can identify which are the actors involved and according to which protocol a negotiation can be executed. As a way to improve the flexibility of the solution accordingly to a SOA, the framework includes the concept of negotiation by delegation. When a negotiation protocol supported by all the participant does not exist, then it is assumed that a participant can fully of partially delegate the negotiation to other services that support negotiation protocols that make it possible to find a common protocol.

Relation to Research Framework: Although the proposed approach is mainly related to the negotiation issues (JRA 1.3), the results can be also exploited to complete the service discovery mechanisms (JRA 2.3). In addition, the flexibility that the approach aims at providing could be exploited for designing adaptable services (JRA 1.2).

Discussion and Future Work: At this stage the framework is mainly focused on the identification of the set of negotiation protocols that can be potentially executed. Future work will be focused on an extension able to select the best negotiation protocols among the discovered ones. In addition, a broader analysis of the impact of this framework in the Service Oriented Architecture has to be analyzed in terms of cost/benefit analysis. Indeed, on the one hand, the proposed framework has the main advantage of making possible the communication of services that could not previously occurred due to the unfeasibility of the negotiation. On the other hand, the effort of describing capabilities and requirements along with the time for executing the discovery process needs to be considered, too.

3.2.18 "High-quality Business Processes based on Multi-dimensional QoS" [34]

Context and Background: The ability of a service end-user (or client application) to take a generic, abstract business process and configure it according to assured non-functional properties (i.e., with a guaranteed quality of service or QoS) is an important principle of service-based application (SBA) instantiation. The same techniques can be used to adapt badly-performing deployed SBAs to meet an expected performance benchmark.

This paper describes three algorithmic techniques for selecting services *differentiated* by their non-functional properties across multiple service *dimensions* to guarantee an end-to-end QoS for an SBA and their evaluation with respect to performance.

Problem Statement: The QoS parameters of a differentiated service may be for any logical level of the service-based application, from the infrastructure layer to business process. Each of these levels is considered a service dimension in the problem at hand. An abstract description of an SBA can be annotated across these dimensions and the whole process with user-specified values (or even with no value indicating no preference on how a particular step should be carried out). This paper presents an algorithmic solution for selecting the concrete services to meet the *constraints* the desired QoS values place on the abstract process.

Relevance of Problem and Progress from State of the Art: Much work exists in the description and matching the functional properties of services, such as through the use of semantic annotations and reasoning over them. Research into the non-functional properties of services has mainly concentrated on the description of those properties and models for considering the selection of services often only consider a single QoS dimension of a particular process, i.e., they only consider a limited search space. This work seeks to extend the state of the art by allowing QoS selection at different abstract levels of a generic business process and to assess the performance of carrying this out over a large search space.

Relation to WP Challenges: This work supports the work in "proactive SLA negotiation and agreement" challenge, as it can be used as a method of selecting different combinations of services as a negotiation progresses. When a mutually acceptable selection is found, an agreement can be formed. The same techniques can be used to re-negotiate an agreement or re-configure a SBA found not to be performing as desired and can be considered as contributing to the "quality prediction techniques to support proactive adaptation". It is a general "run-time quality assurance technique".

Solution / Research Method: The paper proposes the service selection for and configuration of service-based applications can be performed by considering an optimal choice of multi-dimensional QoS variables using the heuristics of simulated annealing and genetic algorithms. The methodology of the work is to use an example SBA and experiment with multiple desired end-to-end QoS and different algorithms to see how they performed, in terms of both accuracy (i.e., the selections difference in end-to-end QoS against what what requested) and time required to complete the selection.

Benefits and Evaluation: The evaluation of the algorithms described in this paper demonstrated that using a generic algorithm gave us a higher accuracy in selecting the services but took much longer than the algorithm based on simulated annealing. A hybrid algorithm that pre-selects services using the simulated annealing algorithm before refining the selection using the genetic algorithm was found to give a slightly higher accuracy than the genetic algorithm but required more time than the genetic algorithm to achieve that result.

Relation to Research Framework: This approach is generic in nature in that it applies equally well to different levels of process granularity, for example, business processes within end-to-end aggregations, component services within individual business processes, or even to activities within component services.

Discussion and Future Work: The results in this paper are preliminary in nature and refinements and extensions are needed in several directions. Initially, we wish to further validate our approach with additional experiments to further calibrate the initial parameters of and the weights attached to the simulated annealing and genetic algorithms in the hybrid algorithm. Following this, we intend to provide integrated support for all constituents of the operational framework we have outlined, and enable the end-user to configure their generic business processes to optimally suit the problem at hand.

3.2.19 "A CMMI Based Configuration Management Framework to Manage the Quality of Service-Based Applications" [21]

Context and Background: Today's computer world consists of applications which are scattered across different networks and require special effort in terms of integration. Software quality assurance is about identifying the right things to implement and test, and allocating and managing resources in a way that minimizes risks when applications and services are deployed. Organizations have business processes in place in order to meet their objectives; e.g., sales, admin, and financial departments work together in a "Sales process". Each of the units involved in an organizational need one or more services (e.g., application software or utilities). These services run on IT infrastructure which includes both hardware and software, it must be managed accordingly to meet organizational objectives.

Problem Statement: Service-Based Applications (SBAs) have highlighted new challenges related to Configuration Management (CM) which is an important process for the assurance of end to end quality in software systems. As far as the quality of SBAs is concerned, configuration management remains an issue because of the loosely coupled and adaptive nature of the corresponding applications. A smart configuration management approach will allow organizations to make their IT resources more reliable and to utilize them to their maximum.

Relevance of Problem and Progress from State of the Art: In the context of enterprise systems to carry out business functions such as sales, administration and financial service, the departments within an organization work together to fulfill business processes. The inter-departmental nature of such processes requires services (e.g., application software or utilities) are shared between business units. These services run on IT infrastructure which includes both hardware and software, therefore it must be managed accordingly to meet the organization's objectives. Correct management of IT infrastructure will ensure that the required services by business processes are available. CM is part of this IT infrastructure which consists of procedures, policies, and documentation.

Relation to WP Challenges: In service-oriented environments the heterogeneity of resources is dealt by virtue of providing any kind of functionality or resource as a service with a stable interface. This however does not completely remove the need for configuration of the resources, which has to be performed by any service provider of an SBA. The configuration of service implementations and resources must therefore comply with the global requirements of the SBA and must be met at the site of each of the distributed pieces of software. Any kind of local configuration management has implications on the service quality and functional properties, which may not be desired.

Solution / Research Method: We propose a CM framework specifically designed with the adaptation of SBAs in mind. The starting point for this framework is the textbook description of the CM process as outlined by Galin. This process description was designed for traditional software development and has many deficiencies in relation to service-oriented development. Galin's CM process description is comprised of a set of configuration activities and their associated action items. Therefore, in order solve this problem, we use practices from CMMI-SVC to support the action items from CM process. As part of our research methodology, we identify the CM activities and all the corresponding action items in them. For each action item, we find corresponding CMMI-SVC practices within its process areas which can support the implementation of these actions items. This one by one mapping of CM supporting action items with relevant CMMI - SVC process areas' practices gives birth to a CM frame work for SBAs. It is worth considering whether or not to use the CMMI-SVC CM process area directly rather than using parts of it as an add-on to reference CM process. However, since CMMI-SVC is targeted at general services and not specifically at Service-Oriented Computing (SOC) or even software, all of its practices are not directly applicable to SOC. Therefore a suitable methodology was to use the skeleton of a Traditional Software Engineering (TSE) CM process and supplement it with software applicable practices from CMMI-SVC.

Benefits and Evaluation: This research proposes a service-based configuration management framework based on SEI-CMMI-SVC which contributes to the S-Cube project. Implementing this approach will allow organizations to effectively manage the configurations of their SBAs. The objective of this research is to create a CM framework that can contribute to the end-to-end quality assurance of SBAs. In terms of end to end quality, a CM process would allow developers more accurately develop and update the correct versions of services within SBAs. Service consumers would also benefit from CM as they would see reorganized services as they are updated correctly.

Relation to Research Framework: In the S-Cube life cycle, the Operations and Management phase belongs to both phases. Therefore, it must be efficient and precise enough to meet the transition needs of the entire life cycle. By improving the CM process within operations and Management, we aim to strengthen the entire life cycle of S-Cube. Existing quality assurance techniques appear to be not yet fully incorporated into a comprehensive life-cycle. These aspects are particularly critical as the designers find that understanding what will happen as a result of some self-adaptation design choice quite difficult. Configuration management is a quality assurance technique, and we are incorporating this into the S-Cube life-cycle under Operations & Management.

Discussion and Future Work: Services have made the world more connected; allowing producers, consumers, and other human resources to communicate frequently across the globe. The service industry is a significant driver for the growth of worldwide economy. So guidance on improving service management procedure can serve as a key contributor to the customer satisfaction, performance, and profitability of the business. In this research, we have proposed a CMMI-SVC based framework to manage the configuration of service-based applications. The hypothesis is supported by means of a case study to depict effectiveness of the approach that business can improve its CM process by means of our contribution. A special case with Service Oriented Architecture is that customers don't see the change of services as long as Service Level Agreements are met. Yet, this is not how it is done nowadays and remains a future research issue for us. Another issue is that sometimes the providers of services in an SBA do not coincide with the SBA provider and they may be discovered dynamically during execution. We intend to use configuration information for the purpose of audit and for ensuring compliance between them.

Chapter 4

Conclusions

To allow service-based applications to be provisioned within short-term, volatile and highly dynamic (business) processes, their end-to-end non-functional properties, or qualities, should be assured so that service consumers can be confident the service-based application will be reliable and meet their requirements.

This deliverable presented 20 contributions that introduce novel and improved (based on validation results) approaches for service quality. Out of the 20 publications (with overlaps),

- 2 focus on quality definition,
- 9 on quality negotiation (including proactiveness), and finally,
- 15 on run-time quality assurance and quality prediction for proactive adaptation.

Clearly, the majority of the presented contributions are concerning principles, techniques and methods for proactive negotiation and adaptation for SBAs. These approaches represent a significant progress towards the main focal point of WP-JRA-1.3, which is on enabling proactive SBAs through exploiting run-time, dynamic quality negotiation and quality prediction techniques. We will summarize and discuss those quality prediction techniques in the following subsection.

The research results of S-Cube concentrating on the quality reference model and on the quality definition language research challenges can be found in the S-Cube deliverables CD-JRA-1.3.2 [18] and CD-JRA-1.3.3 [27], respectively.

4.1 Quality Prediction

We have observed that many researchers within and beyond S-Cube work on the topic of quality prediction. Where there is an agreement on the general motivation and problem, the solutions that are pursued differ. The technology to realize quality prediction that is employed in the proposed solutions include the following ones, as presented in this deliverable:

- *Data Mining (Sections 3.2.1, 3.2.2 and 3.2.3)*: Data mining refers to extracting or "mining" knowledge from large amounts of data. Data mining from previous executions and/or the previous history of the current execution is used to predict future behavior and quality. The proposed solutions include variants of statistical methods as well as machine learning techniques, which rely on the existence of a past history and the assumption that past (or different pasts) can be trusted to predict future behavior.
- *Online Testing (Sections 3.2.6, 3.2.7 and 3.2.8)*: Online testing means that the service-based application is tested (i.e, fed with dedicated test input) in parallel to its normal use and operation [47]. Online testing of constituent services of a service-based application is proposed to actively

collect additional data points to the ones available from monitoring. Thereby, the data points used for prediction are systematically increased – if needed – to improve the precision of the quality prediction.

- *Run-time Verification (Sections 3.2.9 and 3.2.10)*: Run-time verification is a formal analysis technique used to ascertain whether some predefined properties are met at run-time [47]. The proposed solutions include the use of run-time model checking, which is proposed to predict whether, based on the past history of service invocations and the assumptions about future service invocations, it will effectively be possible for the execution of the service-based application to finish successfully. Other solutions propose using soft constraints to model SLAs and to relax those constraints to predict the quality and to decide how to rebuild a composition even in the case that not all the requirements can be satisfied.

- *Static Analysis (Sections 3.2.11, and in CD-JRA-1.3.4 [48] Sections 3.2.5 and 3.2.11)*: Static analysis systematically examines an artifact to determine (infer or "synthesize") certain properties [47]. Those properties can include approximations of the future of the computation. It has been used in the solutions presented in this deliverable to predict at every moment ranges (i.e., upper and lower bounds) for the behavior of the remaining of the computation by using the initial approximation corrected with the actual state.

- *Simulation (Sections 3.2.4 and 3.2.5)*: Dynamic models are executed to simulate the behavior of service-based applications and thus to and predict their future behavior and thus quality properties. For example, orchestrations are transformed into a dynamic model which also models the resources available to execute the orchestration (e.g., number of simultaneous threads). This model, together with assumptions regarding the statistical properties of the components and the input regime of invocations, are subject to simulation which predicts the behavior of the system until a steady state is reached. Alternatively, Discrete-Event-Simulation (DES) is used, which allows the prediction of the performance of service-based applications in different settings and load conditions.

Each of these quality prediction approaches work differently, in different settings and with different assumptions, and at different stages of the life-cycle of a service-based application. Ideally, we would like to be able to choose from the best of all worlds for each situation, and, if possible, to dynamically switch between the quality prediction approaches or jointly use several ones for plausibility checks. To effectively select the best approach for every scenario, the conditions in which each of these approaches can be applied and brings a competitive advantage over the others need to be known.

To work towards the above goal, the members of WP-JRA-1.3, together with external collaborators (especially from SLA@SOI), have initiated a working group on quality prediction. This working group aims at collecting, contrasting and comparing the various quality prediction approaches within and beyond S-Cube. The current results of the working group and a list of its members can be accessed at http://www.s-cube-network.eu/quality-prediction.

4.2 Future Research Activities on Service Quality

As can be seen from the summarized papers and the discussion in the previous section, a very good understanding and a wide range of potential solutions to the problems of proactive negotiation and quality prediction have been achieved. After promising, initial validations of those solutions, the workpackage in the remainder of the project will continue the validation of results to gather a better understanding about the circumstances under which the approaches work well and how they could complement each other. This will lead to CD-JRA-1.3.6: "Validated principles, techniques and methodologies for specifying end-to-end quality and negotiating SLAs for assuring end-to-end quality provision and SLA conformance". The validation of those techniques will also be performed in collaboration with WP-IA-3.2 "Validation of Integration Framework". In collaboration with this workpackage, we will assess the conceptual integration of the approaches by following the validation strategy of exploiting "high-level scenarios along the S-Cube life cycle" (cf. CD-IA-3.2.4).

Bibliography

[1] American customer satisfaction index (acsi). `http://www.theacsi.org/`.

[2] L. An and J.J. Jeng. Web service management using system dynamics. In *ICWS*, pages 347–354. IEEE Computer Society, 2005.

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

[4] A. P. Barros and M. Dumas. The rise of web service ecosystems. *IT Professional*, 8:31–37, September 2006.

[5] S. Benbernou. State of the art report, gap analysis of knowledge on principles, techniques and methodologies for monitoring and adaptation of sbas. Deliverable PO-JRA-1.2.1, S-Cube Consortium, July 2008.

[6] M. Bitsaki, O. Danylevych, W-J. Heuvel, G. Koutras, F. Leymann, M. Mancioppi, C. Nikolaou, and M. Papazoglou. An architecture for managing the lifecycle of business goals for partners in a service network. In *Proceedings of the 1st European Conference on Towards a Service-Based Internet*, ServiceWave '08, pages 196–207, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] M. Bitsaki, O. Danylevych, W-J. A. Heuvel, G. D. Koutras, F. Leymann, M. Mancioppi, C. N. Nikolaou, and M. P. Papazoglou. Service-oriented computing — icsoc 2008 workshops. chapter Model Transformations to Leverage Service Networks, pages 103–117. Springer-Verlag, Berlin, Heidelberg, 2009.

[8] G. Canfora and M. Di Penta. Testing services and service-centric systems: challenges and opportunities. *IT Professional*, 8(2):10–17, 2006.

[9] N. S. Caswell, C. Nikolaou, J. Sairamesh, M. Bitsaki, G. D. Koutras, and G. Iacovidis. Estimating value in service systems: a case study of a repair service system. *IBM Syst. J.*, 47:87–100, January 2008.

[10] M. Comuzzi, K. Kritikos, and P. Plebani. A semantic based framework for supporting negotiation in service oriented architectures. In *Proceeding of 11th IEEE Conference on Commerce and Enterprise Computing (CEC 2009)*. IEEE Computer Society Press, 2009.

[11] R. Dechter. *Constraint Processing*. Morgan Kaufman, 2003.

[12] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K. Pohl. A journey to highly dynamic, self-adaptive service-based applications. *Automated Software Engineering*, 2008.

[13] D. Dranidis, A. Metzger, and D. Kourtesis. Enabling proactive adaptation through just-in-time testing of conversational services. In *Third European Conference, ServiceWave 2010*. Springer, 2010.

[14] D. Dranidis, E. Ramollari, and D. Kourtesis. Run-time verification of behavioural conformance for conversational web services. In *Seventh IEEE European Conference on Web Services (ECOWS 2009), 9-11 November 2009, Eindhoven, The Netherlands*, pages 139–147, 2009.

[15] J. Ejarque, A. Micsik, R. Sirvent, P. Pallinger, L. Kovacs, and R. M. Badia. Semantic resource allocation with historical data based predictions. In *The First International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2010*, 2010.

[16] J. Esparza and C. Lakos, editors. *Applications and Theory of Petri Nets 2002*, volume 2360 of *Lecture Notes in Computer Science*. Springer Verlag, 2002.

[17] A. Gehlert, A. Bucchiarone, R. Kazhamiakin, A. Metzger, M. Pistore, and K. Pohl. Exploiting assumption-based verification for the adaptation of service-based applications. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 2430–2437, New York, NY, USA, 2010. ACM.

[18] A. Gehlert and A. Metzger. Quality reference model for SBA. Deliverable CD-JRA-1.3.2, S-Cube Consortium, 2008.

[19] C. Ghezzi and S. Guinea. Run-time monitoring in service-oriented architectures. In Luciano Baresi and Elisabetta Di Nitto, editors, *Test and Analysis of Web Services*. Springer, 2007.

[20] R. Hamadi and B. Benatallah. A petri net-based model for web service composition. In *ADC '03: Proceedings of the 14th Australasian database conference*, pages 191–200, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[21] S.I. Hashmi, S. Lane, D. Karastoyanova, and I. Richardson. A cmmi based configuration management framework to manage the quality of service based applications. In *EuroSPI: Industrial Proceedings of the 17th European conference on Software Process Improvement*, 2010.

[22] J. Hielscher, R. Kazhamiakin, A. Metzger, and M. Pistore. A framework for proactive self-adaptation of service-based applications based on online testing. In *ServiceWave 2008*, number 5377 in LNCS. Springer, 10-13 December 2008.

[23] W. Hummer, O. Raz, O. Shehory, P. Leitner, and S. Dustdar. Test coverage of data-centric dynamic compositions in service-based systems. In *In Proceedings of the 4th International Conference on Software Testing, Verification and Validation (ICST)*, 2011.

[24] D. Ivanović, M. Carro, and M. Hermenegildo. 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, 5-10 July 2010*. IEEE, 2010.

[25] D. Ivanović, M. Treiber, M. Carro, and S. Dustdar. Building dynamic models of service compositions with simulation of provision resources. In *Proceedings of the 29th international conference on Conceptual modeling*, ER'10, pages 288–301, Berlin, Heidelberg, 2010. Springer-Verlag.

[26] Y. Jamoussi, M. Driss, J.-M. Jézéquel, and H. Ben Ghézala. Qos assurance for service-based applications using discrete-event simulation. *IJCSI International Journal of Computer Science Issues*, 7(4), 2010.

[27] K. Kritikos and B. Pernici. Initial concepts for specifying end-to-end quality characteristics and negotiating slas. Deliverable CD-JRA-1.3.3, S-Cube Consortium, June 2009.

[28] K. Kritikos, B. Pernici, P. Plebani, C. Cappiello, M. Comuzzi, S. Berbernou, I. Brandic, A. Kertész, M. Parkin, and M. Carro. A survey on Service Quality Description. submitted to ACM Computing Surveys, 2011.

[29] D. D. Lamanna, J. Skene, and W. Emmerich. SLAng: A Language for Defining Service Level Agreements. In *FTDCS 2003: Proceedings of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 100–107. IEEE Computer Society, May 2003.

[30] J. H. Lee, Y. S. Han, and C. H. Kim. It service management case based simulation analysis and design: Systems dynamics approach. In *Convergence Information Technology, 2007. International Conference on*, pages 1559 –1566, 21-23 2007.

[31] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar. Monitoring, Prediction and Prevention of SLA Violations in Composite Services. In *Proceedings of the 2010 IEEE International Conference on Web Services (ICWS'10)*, 2010.

[32] P. Leitner, B. Wetzstein, D. Karastoyanova, W. Hummer, S. Dustdar, and F. Leymann. Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution. In *Proceedings of the International Conference on Service-Oriented Computing (ICSOC'10), to appear.*, 2010.

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

[34] Q. Liang and M. Parkin. *Performance and Dependability in Service Computing: Concepts, Techniques and esearch Directions*, chapter High-quality Business Processes based on Multi-dimensional QoS. IGI Global, 2011. to appear.

[35] H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck. Web Service Level Agreement (WSLA) Language Specification. Document wsla-2003/01/28, IBM Corporation, 2003.

[36] K. Mahbub and G. Spanoudakis. A framework for proactive sla negotiation. In *5th International Conference on Software and Data Technologies (ICSOFT 2010), 22 – 24 July, Athens, Greece*, 2010.

[37] K. Mahbub and G. Spanoudakis. Proactive sla negotiation for service based systems. In *Proceedings of the 2010 IEEE World Congress on Services (SERVICES 2010), Miami, Florida, USA, 5-10 July*, 2010.

[38] A. Metzger, O. Sammodi, K. Pohl, and M. Rzepka. Towards pro-active adaptation with confidence: Augmenting service monitoring with online testing. In *Proceedings of the ICSE 2010 Workshop on Software Engineering for Adaptive and Self-managing Systems (SEAMS 2010)*, Cape Town, South Africa, 2010.

[39] A. Metzger, E. Schmieders, C. Cappiello, E. Di Nitto, R. Kazhamiakin, B. Pernici, and M. Pistore. Towards proactive adaptation: A journey along the s-cube service life-cycle. In *MESOA: 4th International Workshop on Maintenance and Evolution of Service-Oriented Systems*, 2010. to appear.

[40] H. Meziane and S. Benbernou. A dynamic privacy model for web services. *Comput. Stand. Interfaces*, 32:288–304, October 2010.

[41] A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar. Comprehensive qos monitoring of web services and event-based sla violation detection. In *MWSOC '09: Proceedings of the 4th International Workshop on Middleware for Service Oriented Computing*, pages 1–6, New York, NY, USA, 2009. ACM.

[42] A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar. End-to-End Support for QoS-Aware Service Selection, Binding, and Mediation in VRESCo. *IEEE Transactions on Services Computing*, 3:193–205, July 2010.

[43] C. Miller, A. Ruiz-Cortes, and M. Resinas. An Initial Approach to Explaining SLA Inconsistencies. In *Service-oriented Computing-Icsoc 2008: 6th International Conference, Sydney, Australia, December 1-5, 2008, Proceedings*, page 394. Springer-Verlag New York Inc, 2008.

[44] U. Montanari. Networks of Constraints: Fundamental Properties and Application to Picture Processing. *Information Sciences 7*, pages 95 – 132, 1974.

[45] OOPSLA Program Committee. How to Get Your Paper Accepted at OOPSLA. In *OOPSLA '91 Proceedings Appendix*, pages 359–363, 1991.

[46] E. Orta, M. Ruiz, and M. Toro. A system dynamics approach to web service capacity management. *Web Services, European Conference on*, 0:109–117, 2009.

[47] M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger, editors. *Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Mechanisms and Methods for Engineering, Managing, and Adapting Service-Based Systems*. Springer, Heidelberg, Germany, 2010.

[48] M. Parkin and A. Metzger. Initial set of principles, techniques and methodologies for assuring end-to-end quality and monitoring of slas. Deliverable CD-JRA-1.3.4, S-Cube Consortium, March 2010.

[49] A. Paschke. RBSLA: A declarative Rule-based Service Level Agreement Language based on RuleML. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06)*, volume 2, pages 308–314. IEEE Computer Society, November 2005.

[50] P. R. Krishna, K. Karlapalem, and D.K.W. Chiu. An ER^{EC} framework for e-contract modeling, enactment, and monitoring. *Data Knowl. Eng.*, 51:31–58, 2004.

[51] S-Cube Partners. Quality Reference Model for SBAs. Deliverable CD-JRA-1.3.2, S-Cube Network of Excellence, March 2008.

[52] A. Sahai, A. Durante, and V. Machiraju. Towards Automated SLA Management for Web Services. Report HPL-2001-310R1, HP Laboratories, Palo Alto, California, July 2002.

[53] O. Sammodi, A. Metzger, X. Franch, J. Marco M. Orio1, and K. Pohl. Usage-based online testing for proactive adaptation of service-based applications. Submitted to COMPSAC 2011.

[54] J. D. Sterman. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin McGraw-Hill, 2000.

[55] T. Andrews and F. Curbera and H. Dholakia and Y. Goland and J. Klein and F. Leymann and K. Liu and D. Roller and D. Smith and S. Thatte and I. Trickovic and S. Weerawarana. Business Process Execution Language for Web Services, 2003.

[56] G. Tselentis, J. Domingue, A. Galis, A. Gavras, and D. Hausheer. *Towards the Future Internet: A European Research Perspective*. IOS Press, Amsterdam, The Netherlands, 2009.

[57] W. van der Aalst. Don't Go With the Flow: Web Services Composition Standards Exposed. *IEEE Intelligent Systems*, Jan/Feb 2003.

[58] W. M. P. van der Aalst and A. H. M. ter Hofstede. Yawl: yet another workflow language. *Inf. Syst.*, 30(4):245–275, 2005.

[59] W.M.P. van der Aalst. Pi calculus versus petri nets: Let us eat ”humble pie” rather than further inflate the ”pi hype”, 2003.

[60] J. Vonk and P. Grefen. Cross-organizational transaction support for E-services in virtual enterprises. *Distrib. Parallel. Dat.*, 14:137–172, 2003.

[61] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson. *Web Services Platform Architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Prentice Hall PTR, 2005.

[62] K. Zachos, C. Nikolaou, P. Petridis, G. Stratakis, M. Voskakis, and E. Papathanasiou. Enhancing service network analysis and service selection using requirements-based service discovery. In *Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD ’09.*, pages 325 –331, 2009.

[63] M.A. Zemni, S. Benbernou, and M. Carro. A soft constraint-based approach to qos-aware service selection. In M. Weske, J. Yang, P. Maglio, and M. Fantinato, editors, *ICSOC 2010*, Lecture Notes in Computer Science. Springer Verlag, 2010.

[64] H. Zhao and H. Tong. A dynamic service composition model based on constraints. In *GCC*, pages 659–662, 2007.