Grant Agreement N° 215483

Title: *Validated principles, techniques and methodologies for specifying end-to-end quality and negotiating SLAs and for assuring end-to-end quality provision and SLA conformance (incl. proactiveness)*

Authors: *CITY, FBK, LERO, POLIMI, UniDue, UPC, UPM*

Editor: *Pierluigi Plebani (POLIMI), Andreas Metzger (UniDue), Osama Sammodi (UniDue)*

Reviewers: *Kyriakos Kritikos (UoC), Gabriele Tolomei (CNR)*

Identifier: *CD-JRA-1.3.6*

Type: *Contractual Deliverable (CD)*

Version: *1.0*

Date: *February 29th, 2012*

Status: *Final*

Class: *External*

Management summary

The aim of this deliverable is to report the results of the fourth year of the S-Cube network on research topics related to proactive quality negotiation and assurance. This paper-based deliverable summarizes the network's research results that have been published in books, journals, and conference proceedings.

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

1. To define principles and techniques for *specifying, negotiating and assuring end-to-end quality provision and SLA conformance (including proactiveness)* with respect to quality characteristics across the functional layers of service infrastructure, service composition and coordination, and business process management, and, across the chain of service providers and consumers. The quality characteristics to be considered included characteristics such as performance, dependability, reliability and availability.

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 (service-based application) engineering framework and the SBA monitoring and adaptation framework.

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

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

As part of these long-term goals, this deliverable provides a consolidated report on results for run-time quality assurance, quality prediction (to enable proactive adaptation) and automated and proactive negotiation, many of which have been experimentally validated, 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

1 Deliverable Overview **3**
- 1.1 Introduction 3
- 1.2 Deliverable Structure 4

2 Research Challenges and Contribution to the Integrated Research Framework **5**
- 2.1 Key Research Challenges 5
  - 2.1.1 Challenges In Quality Definition 6
  - 2.1.2 Challenges In Quality Negotiation 7
  - 2.1.3 Challenges in Quality Assurance 7
- 2.2 Contribution of Deliverable to Key Research Challenges 8

3 Principles, Techniques and Methodologies for Service Quality **9**
- 3.1 Structured Presentation of Results 9
- 3.2 Summary of Research Results 10
  - 3.2.1 Usage-Based Online Testing for Proactive Adaptation of Service-Based Applications 10
  - 3.2.2 Towards Accurate Failure Prediction for the Proactive Adaptation of Service-oriented Systems 13
  - 3.2.3 Accurate Service Failure Prediction through Online Testing 15
  - 3.2.4 Proactive SLA Negotiation for Service Based Systems: Initial Implementation and Evaluation Experience 17
  - 3.2.5 SALMonADA: A platform for Monitoring and Explaining Violations of WSAgreementcompliant Documents 19
  - 3.2.6 Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification 21
  - 3.2.7 Future Internet Apps: The Next Wave of Adaptive Service-Oriented Systems? . . 22
  - 3.2.8 Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology 23
  - 3.2.9 SLAs for Cross-layer Adaptation and Monitoring of Service-Based Applications: A Case Study 25
  - 3.2.10 Negotiation towards Service Level Agreements: A Life Cycle Based Approach . 27
  - 3.2.11 A Context-Aware Framework for Business Process Evolution 28
  - 3.2.12 Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations 32

4 Conclusions **35**
- 4.1 Future Research Activities on Online Quality Prediction 35

Chapter 1

Deliverable Overview

1.1 Introduction

Although work in Service Oriented Computing (SOC) often focuses on the what a service is able to perform, i.e., its functional aspects, quality (aka. non-functional) aspects play an important role during the entire life-cycle of a service-based application (SBA). As an example, when services are designed and deployed, their performance needs to be taken into account to ensure their usefulness. In this way, during the service discovery and matchmaking phase, service integrators (who build and deploy the SBA), when given a set of functionally equivalent services, are able to select which is the best amongst them. Employing suitable negotiation frameworks, such service selection can be personalized in a sense that the service provider and the service integrator (consumer) can "make a deal" concerning the behaviour of the services when they will be invoked. Ultimately, this leads to contracts (aka. Service-Level Agreements – SLAs) being required in the setting of SBAs. Finally, when the invocation takes place, both parties need to be sure that the execution satisfies what was established in the contract, i.e., they need to perform quality assurance of the services.

The work performed in workpackage WP-JRA-1.3 during the whole duration of the S-Cube network focused on all these aspects that, according to [37, 50], can be summarized as follows:

- *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., an 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*: The 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 requirements. 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 summarizes the results achieved during the fourth year of the S-Cube network, especially in terms of quality negotiation and assurance considering the end-to-end quality of SBAs. This deliverable documents principles and techniques for automated and proactive quality assurance and negotiation. The work concerns (1) handling quality of SBAs that are composed by services that may be provided by different third-party providers, (2) run-time prediction of the quality provided by the whole application as well as its services, (3) techniques to support the proactive adaptation of the application in case imminent failures may affect the quality of the overall application.

1.2 Deliverable Structure

As this deliverable is paper-based, it contains two parts: (a) This document, which provides the overall motivation and summary of the key research outcomes of the workpackage. (b) 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 Chapter 2, this deliverable provides an overview of the research challenges being pursued within WP-JRA-1.3. In Chapter 3, the document – based on research publications of S-Cube members and associate members – reports on a set of principles and techniques for quality definition, negotiation and assurance developed in year 4 of the network, thereby contributing to the research challenges of the workpackage. Chapter 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. Those challenges aimed at addressing key research gaps identified in [32]. Thereby, this chapter allows for relating the results presented in this deliverable to the research objectives of the workpackage (see Sections 2.2 and Chapter3). Specifically for year 4 of the project, focus of research activities was on challenges related to "proactiveness", i.e., on challenges introduced in Sections 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.

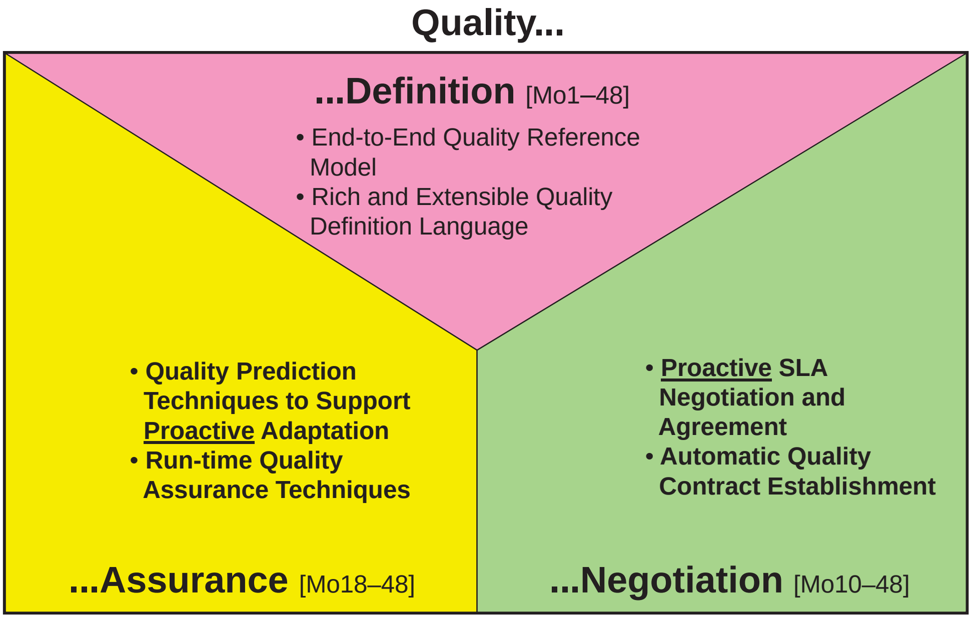

Figure 2.1: Activities relevant for quality of SBAs and key research objectives of workpackage

2.1.1 Challenges In Quality Definition

Note: Work on challenges related to quality definition constituted the foundations for the work on other challenges of the workpackage. Those activities have been finalized – as planned – by year 2 of S-Cube. The research results related to these research challenges can be found in the S-Cube deliverables CD-JRA-1.3.2 [9] and CD-JRA-1.3.3 [19].

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 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.

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 [42]. 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 [23], WSML [43], SLAng [20] and RBSLA [34] (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.

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 solutions, self-adaptation often happens after a change or a deviation has occurred, i.e., in a reactive fashion. 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 be observed by the users of the SBA. Key to proactive adaptation is to predict the future quality (and functionality) of a SBA and to proactively plan mitigation and repair activities [44] 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. These build on concepts such as online testing, run-time model analysis, model-checking at run-time, static analysis, simulation and machine learning.

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 and techniques for defining, negotiating and assuring end-to-end quality for SBAs.

This deliverable provides an overview of the research results of the WP-JRA-1.3 members for year 4 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 contributions 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" [31], 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 achieved by the S-Cube project during the fourth year in the areas of Quality Negotiation, and Assurance. The research results (i.e., papers) are categorized in relation to their contribution to the research challenges of this workpackage, described earlier in Sections 2.1.2–2.1.3. In accordance to the objective of the deliverable, the research results mainly focus on the proactive quality negotiation and assurance. Giving to the SBA the capability of an automatic negotiation and to estimate possible future failures is an important enabling factors for the SBA adaptation. For this reason many results presented here are also tightly related to research challenges pursued in WP-JRA-1.2 (Monitoring and Adaptation).

Sections 3.2.1–3.2.12 now provide the descriptions of the research results according to the standard structure described above.

3.2.1 "Usage-Based Online Testing for Proactive Adaptation of Service-Based Applications" [45]

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 system's 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 provider's 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

Section	Paper Title [reference]		Quality Negotiation: Automatic Contract Establishment	Quality Negotiation: Proactive SLA Negotiation & Agreement	Quality Assurance: Run-time Quality Assurance	Quality Assurance: Quality Prediction for Proactive Adaptation	Also Contributes to Workpackage
3.2.1	(+) Usage-Based Online Testing for Proactive Adaptation of Service-Based Applications	[45]			✓	✓	JRA-1.2, JRA-2.2
3.2.2	(+) Towards Accurate Failure Prediction for the Proactive Adaptation of Service-oriented Systems	[27]			✓	✓	JRA-1.2, JRA-2.2
3.2.3	(*) Accurate Service Failure Prediction through Online Testing	[46]			✓	✓	JRA-1.2, JRA-2.2
3.2.4	(+) Proactive SLA Negotiation for Service Based Systems: Initial Implementation and Evaluation Experience	[25]		✓			JRA-1.2
3.2.5	(*) SALMonADA: A platform for Monitoring and Explaining Violations of WSAgreementcompliant Documents	[24]	✓	✓	✓	✓	JRA-1.2
3.2.6	(+) Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification	[47]			✓	✓	JRA-1.1
3.2.7	(+) Future Internet Apps: The Next Wave of Adaptive Service-Oriented Systems?	[29]			✓	✓	JRA-1, JRA-2
3.2.8	(+) Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology	[26]			✓	✓	JRA-1.2, JRA-2.2
3.2.9	(+) SLAs for Cross-layer Adaptation and Monitoring of Service-Based Applications: A Case Study	[5]			✓	✓	JRA-1.1, JRA-1.2, JRA-2.1, JRA-2.2, JRA-2.3
3.2.10	(+) Negotiation towards Service Level Agreements: A Life Cycle Based Approach	[13]		✓			JRA-1.1, JRA-1.2
3.2.11	(+) A Context-Aware Framework for Business Process Evolution	[6]				✓	JRA-1.1, JRA-1.2, JRA-2.1
3.2.12	(+) Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations	[16]				✓	JRA-2.1, JRA-2.2

Table 3.1: Coverage of Research Challenges by Research Results ((*) = submitted, (+) = accepted)

need for adaptation 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 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: This work addresses the research challenges 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 work we address the above problems and introduce a novel online testing technique that provides enhanced proactive adaptation capabilities to SBAs. More specifically, the work provides the following major contributions:

- A framework for proactive adaptation, which exploits synergies between monitoring, online testing and quality prediction. The framework's core element is a test selection activity that utilizes information about the usage of the SBA's 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 the proposed 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, in order to evaluate and analyze 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 (in 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 to Monitoring and Adaptation (JRA-1.2) as the combined monitoring and testing data is used for making precise failure prediction to support proactive adaptation.

Discussion and Future Work: A quality prediction framework for proactive adaptation of SBAs that exploits usage-based online testing has been presented. 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 an SBA's 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 required 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.

3.2.2 "Towards Accurate Failure Prediction for the Proactive Adaptation of Service-oriented Systems" [27]

Context and Background: Service-orientation is increasingly adopted as a paradigm for building highly dynamic, distributed service-oriented systems. A service-oriented system is realized by composing individual software services. In contrast to a software component, not only the development, quality assurance, and maintenance of the software can be under the control of third-parties, but the software can also be executed and managed by third-parties.

There is a clear trend that in the emerging "Future Internet", service-oriented applications will be increasingly composed of third-party services accessible over the network. As a consequence, the capabilities and quality of service-oriented systems more and more will depend on the quality of their third-party services.

In particular, this means that service-oriented systems will need to become resilient against failures of their third-party services. As a simple example, a service might become unavailable due to overloads on the service provider's side. Furnishing service-oriented systems with self-adaptation capabilities is considered a key solution to address this challenge.

Problem Statement: To trigger the proactive adaptation of a service-oriented system, pending failures need to be predicted. It is important that such a failure prediction is accurate, such as to avoid the execution of unnecessary proactive adaptations, as well as not to miss proactive adaptation opportunities.

Unnecessary adaptations can have the following severe shortcomings: Firstly, unnecessary adaptations can be costly. 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 service-oriented system, e.g., if a seemingly unreliable but cheap service is replaced by a more costly one. Secondly, unnecessary adaptations could be faulty (e.g., if the new service has bugs), leading to severe problems as a consequence. Thirdly, as executing the adaptation takes time, this means that in the worst case, an unnecessary adaptation will leave less time to address actual failures.

In case an adaptation opportunity is missed due to inaccurate failure predictions, this obviously can lead to the same shortcomings as faced in the setting of reactive adaptations, i.e., it can require compensation or costly repair activities. This means, inaccurate predictions would diminish the benefits of proactive adaptation.

Providing accurate failure predictions is extremely challenging in the setting of service-oriented systems, if third-party services are present. The observed quality and functionality of those third-party services can significantly vary between different service invocations. For instance, the performance of a third-party service might depend on the load of the infrastructure at the provider's side or the network latency, if services are offered over the Internet. As an example, a failure observed at one point in time (e.g., unavailability of a service because of an overload at the service provider side) can disappear at a later point in time (e.g., the same service is now executed because of a lower load at the service provider side).

Relevance of Problem and Progress from State of the Art: Several quality prediction approaches exist in the literature. Each approach works differently, in different settings and with different assumptions; e.g., machine learning works well once a significant amount of training data has been collected. Still, all of those techniques share the same concern: they need to accurately predict the future quality, resp. future failures, of a service-oriented system.

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

Solution / Research Method: This work introduces two directions along which accurate failure predictions for proactive adaptation could be established; firstly, by improving the prediction techniques themselves; secondly, by dynamically estimating the accuracy during run-time. Based on selected prediction techniques from the literature and metrics to assess the accuracy of predictions, those two directions for achieving accuracy are critically discussed.

Benefits and Evaluation: The discussions in this work are backed by results from experiments which are based on: (1) simulation of an example service-oriented system and its associated third-party services, together with (2) a prototypical implementation of selected quality prediction techniques.

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 to Monitoring and Adaptation (JRA-1.2), as the combined monitoring and testing data is used for making precise failure prediction to support proactive adaptation.

Discussion and Future Work: Failure prediction (or quality prediction) techniques are key to engineer service-oriented systems with proactive adaptation capabilities. However, as discussed previously, those predictions have to be accurate, as – for instance – false predictions can lead to additional operational costs and severe failures.

Research in the field has produced a diverse range of prediction techniques that are applicable in different settings and have various benefits and shortcomings. Of course, one could strive to design techniques that provide high accuracy for a known setting.

However, if we consider the highly dynamic nature of service-oriented systems in the "Future Internet", even a failure prediction technique which provided good accuracy for known settings can quickly become "obsolete", as it will be highly probable that new, unforeseen settings will dynamically arise.

Ultimately, this suggests to further investigate quality prediction techniques that also provide capabilities to dynamically assess their accuracy during runtime.

3.2.3 "Accurate Service Failure Prediction through Online Testing" [46]

Context and Background: Web-based services provide unprecedented opportunities to build highly flexible systems by integrating service offerings from third parties. However, service consumers and integrators have limited control over third-party web-based services. Those services thus may behave in ways not anticipated during design time, leading to failures during run-time. For example, those services may exhibit a degradation of quality of service (QoS), such as reduced performance or low reliability.

Online failure prediction of third-party services allows anticipating degradations in expected QoS (e.g., as stipulated in SLAs). Online failure prediction thus allows, for instance, to plan and implement proactive repair or compensation activities. Several online failure prediction techniques have been presented in literature. All these techniques all rely on monitoring of QoS data to predict failures.

Monitoring only passively observes services during their actual use. The amount and timeliness of QoS data collected by monitoring thus may be limited; e.g., when a service is only seldom invoked by users. Sparse monitoring data may undermine the accuracy of failure predictions. On the one hand, this may lead to false positive predictions, which imply unnecessarily switching to an expensive service, delays due to unnecessary repair activities, or replacing a working service with another one that has severe defects. On the other hand, this may lead to false negative predictions, which mean that the opportunity for proactive repair and compensation may be missed altogether.

Problem Statement: Online testing has been proposed as an active quality assurance technique to complement passive monitoring. Online testing means that constituent services of a service-oriented system are systematically tested in parallel to its normal use. QoS data of tested services can thus be used to augment QoS data from monitoring. Our previous work indicates that such augmented data improves accuracy of failure predictions of web-based services.

However, online testing implies additional costs for operating software systems; e.g., when online testing a pay-per-use service. To become applicable in practice, one has to understand when online testing pays off with respect to improvements in accuracy gains.

Different factors can impact accuracy gains achieved by online testing. For example, gains in accuracy diminish as more frequent monitoring data is available. After a certain point, online testing thus may not pay off.

Relevance of Problem and Progress from State of the Art: In general, failure prediction approaches rely on QoS data from monitoring. However, due its "passive" nature, the amount and timeliness of the collected data may be limited, which may undermine the accuracy of failure predictions. Our proposed framework , which is called PROSA, complements such approaches by collecting timely QoS data using online testing.

In the literature, several techniques and platforms exist that advocate to perform online testing periodically or event-driven, but only for runtime quality assurance. However, exploiting online testing for online failure prediction has not been addressed.

For combining online testing with monitoring, two main directions can be observed in the literature. Firstly, using monitoring data to build the usage profiles or usage models. However, the direction was proposed for test case definition or offline testing and not for triggering online tests.

Secondly, monitoring data was exploited to reduce the number of service invocations when executing a test suite (i.e., reducing the cost of testing). Approaches used this direction to mimic the service responses. However, they do not complement monitoring data with online testing for failure prediction.

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

Solution / Research Method: This work presents the setup and results of extensive experiments to empirically assess how accuracy gains achieved by online testing depend on different factors, including online test rates, usage frequencies, failure prediction models, and failure rates. The experiments are based on QoS data of real-world web-based services.

Based on the experimental findings, this work introduces an extension of an existing Web Services monitoring framework. The extended framework can be parameterized such that online tests are only triggered when the costs of online testing may pay off with respect to accuracy gains. The conceptual design of the framework and the technical implementation of its extended monitoring engine, are described in this work.

Benefits and Evaluation: As already mentioned, the influential factors are analyzed based on extensive experiments using real Web services. The results clearly indicate that service usage frequency, online test rates, prediction model, and service failure rates have clear influence on the accuracy gains achieved by online testing.

Consequently, this work presents a framework which can be parameterized to trigger online tests if the costs of online testing may pay off with respect to accuracy gains in failure predictions.

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 to Monitoring and Adaptation (JRA-1.2), as the combined monitoring and testing data is used for making precise failure prediction to support proactive adaptation.

Discussion and Future Work: Online testing promises to improve the accuracy of online failure prediction by complementing passive monitoring. However, online testing can incur additional costs for operating software systems. Understanding when online testing pays off with respect to accuracy gains is thus an important issue for the applicability of online testing. To this end, this work has presented the results of extensive experiments conducted to empirically assess how accuracy gains from online testing depend on different factors, including online test rates, usage frequencies, and failure prediction models. Based on the experimental findings, an extension of an existing monitoring framework for Web Services was introduced, which is called PROSA. PROSA can be parameterized to trigger online tests when the costs of online testing may pay off (depending on the application settings) with respect to accuracy gains in failure predictions.

Future work includes using PROSA to complement approaches for predicting failures for a service-oriented system or a composite service.

3.2.4 "Proactive SLA Negotiation for Service Based Systems: Initial Implementation and Evaluation Experience" [25]

Context and Background: Service Level Agreements (SLA) define quality of service (QoS) and functional properties, which should be guaranteed during the provision of a software service, as well as the penalties that should be applied in case the properties are not fulfilled. An SLA is set through a negotiation between the provider and the consumer of a service. SLA negotiation can be particularly complex depending on the requirements and affordances of the two parties. Furthermore, it may need to be carried out at runtime, if a constituent service of a service based system (SBS) becomes unavailable whilst SBS is in operation, or it fails to perform according to its established SLA. There are a number of possible scenarios that may lead to the violation of an SLA. More specifically, a SLA may be violated due to, i) poor QoS 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 and v) change of provider's circumstances, e.g. provider suffers from peak demand of services from its requesters. In such cases, an SBS should be able to discover alternative replacement services for the failed service, and negotiate SLAs with their providers at runtime.

Problem Statement: To minimize the runtime interruption of the SBS, the discovery of replacement services for the SBS's constituent ones should be proactive, i.e., it should be performed before a constituent service of SBS becomes unavailable or fails to perform according to its established SLA. Proactiveness is important since service discovery is a time consuming activity and, therefore, carrying it in a reactive mode, is likely to cause significant interruption in the provision of the composite service and violations of its own SLA. SLA negotiation should also be proactive, as it will be necessary to have adequate SLAs for the potential replacement services that have been identified by proactive discovery, while attempting SLA negotiation just prior to binding to an alternative service is likely to cause significant delay.

Relevance of Problem and Progress from State of the Art: Existing work on service level agreements has focused on SLA specification, negotiation and monitoring. The need for runtime SLA negotiation or re-negotiation has also been addressed in the literature, where either the terms of an SLA are revised to accept a constituent service from an existing provider or a new SLA is negotiated with a new service provider and an existing SLA is terminated. All these approaches, however, are reactive as they support corrective actions only after an SLA has been violated. Thus they may fail to guarantee uninterrupted runtime provision of composite services.
To address the above shortcomings, a framework was developed for integrating proactive SLA negotiation with dynamic service discovery in order to provide cohesive runtime support for both these activities. The proactive negotiation of SLAs, as part of service discovery is necessary for reducing the extent of interruptions during the operation of a SBS, when the need for replacing services in SBS arises.

Relation to WP Challenges: This work addresses the research challenge *Proactive SLA Negotiation and Agreement* and also partially addresses the challenges *Adaptation and Monitoring* and *Discovery and Registry Infrastructure* (defined in WP-JRA-2.3) as we argue about the combination of service discovery and monitoring to facilitate the proactive SLA negotiation.

Solution Research Method: In this work we have produced a framework called PROSDIN (PROactive Service DIscovery and Negotiation). In this framework, we have developed a proactive runtime SLA negotiation tool, and integrated it with a tool supporting proactive runtime service discovery. More specifically, in PROSDIN, SLA negotiation has been developed as an integrated part of the service discovery process, enabling the execution of both activities in a coordinated manner. Proactive SLA negotiation is performed immediately after the execution of service discovery queries to ensure that adequate SLAs are provisionally agreed for given periods of time with the providers of the discovered services, if possible. Also when a pre-agreed SLA expires, it is proactively re-negotiated.
The service discovery tool in PROSDIN is used to identify candidate services that could potentially be used by the SBS. Service discovery is based on queries that express conditions about the interface, behaviour, contextual and quality characteristics of services. To use PROSDIN, each of the constituent SBS services, which is replaceable at runtime, should be associated with a discovery query which, specifies the conditions for discovering services that could potentially replace this constituent service. These queries should be specified by the developer of the SBS during the SBS development, and passed to PROSDIN by the SBS at runtime in order to be executed when service failures occur and enable service discovery. Following the subscription of such queries for a constituent service S_c of an SBS, PROSDIN executes them proactively and in parallel with the execution of the SBS, and stores the services that match them in an external registry to maintain an up-to-date set of candidate replacement services for S_c.
The negotiation tool in the PROSDIN manages the negotiation process on behalf of service client applications. In particular, it provides access to different negotiation engines that may be plugged into the framework by translating negotiation rules expressed in the common language of the framework, into the different negotiation specifications accepted by these engines, and realizes the interface for interacting with broker which, carry out the negotiation process on behalf of services. These brokers may be the same as the broker used by PROSDIN or other brokers that realize the same SLA negotiation interaction interface with it. The negotiation process is carried out according to a two-phase protocol that may result either in a pre-agreed but not activated SLA or fail. Pre-agreed SLAs have an expiry period within which they can become active, if the service client application decides to activate them.

Benefits and Evaluation: We performed a series of experiments to evaluate the implementation of the PROSDIN framework. The purpose of these experiments was to: (a) measure the overhead of SLA negotiation (whether reactive or proactive) on the execution time of the runtime service discovery process, and (b) assess the effectiveness of proactive SLA negotiation over reactive SLA negotiation during runtime service discovery process. In the experiments, we have used an SBS implemented as a BPEL service orchestration process. The service discovery query used in the experiments was specified in order to identify candidate replacement services for a constituent service of the SBS. In the experiments, we also used an SLA template with four QoS terms for negotiation. For negotiation, we specified a set of 15 service consumer negotiation rules (CNR set), and 20 different sets of provider negotiation rules (PNR sets). During negotiation with each of the candidate services identified by the discovery process, the negotiation broker of the service provider side picked up randomly one of the PNR sets and carried out the negotiation based on it. To assess whether the number of considered services affects the performance of the service discovery and SLA negotiation processes, we performed the experiments with three different service sets (registries). These sets contained 100, 300 and 500 services, respectively, and were populated with appropriate service specifications.
As shown in Table 3.2, the time required to select a replacement service in case of service discovery with proactive SLA negotiation is slightly larger than the time required to identify a replacement service in case of service discovery without any SLA negotiation is used. This is because in the former case, the pre-agreed SLA needs to be activated before the replacement service is returned. It should be noted, however, that the main benefit shown in the table is that the time required to select and bind a replacement service at runtime in the case of service discovery with reactive SLA negotiation is significantly larger than the service selection and binding time in the case of service discovery with proactive SLA

Table 3.2: Proactive SLA Negotiation Evaluation Results

	SD Only			SD with Proactive SLA			SD with Reactive SLA		
	100	300	500	100	300	500	100	300	500
Replacement Service Set Maintenance (avg time in ms)	690.5	698.3	673.5	837.2	881.2	892	637.4	668.5	677.6
Selection of Replacement Service (avg time in ms)	22	28	21.8	53.2	45.8	50.2	453	459.4	443.8

negotiation.

Relation to Research Framework: The approach presented in this paper focuses on the proactive SLA negotiation (JRA-1.3). Moreover, this approach performs service discovery (JRA-2.3) and the monitoring of agreed SLAs to enable proactive negotiation that facilitates the proactive adaptation of a service based system (JRA-1.2).

Discussion and Future Work: The objective of proactive SLA negotiation in PROSDIN is to ensure that a service, which could be potentially used by a service client application, will have an agreed set of guaranteed provision terms, if the need to deploy it arises at runtime. Hence, when this need arises, it will not be necessary to engage in a lengthy negotiation process interrupting the operation of the service client application. Our approach has been evaluated through an initial set of experiments showing that proactive SLA negotiation leads to significant reduction of the time required to perform service replacement at runtime, if the existence of agreed SLAs is a prerequisite for service use. PROSDIN opens a spectrum of possible lines for future investigation. These include support for proactive negotiation of hierarchical SLAs, i.e., SLAs of complex composite services deploying other composite services with their own sub-SLAs which will need to be negotiated separately and before coming to a higher level service level agreement. Also 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.

3.2.5 "SALMonADA: A platform for Monitoring and Explaining Violations of WSAgreementcompliant Documents" [24]

Context and Background: Service Level Agreements (SLAs) establish the service quality between consumers and providers of service-based systems (SBS), so there is a need for monitoring techniques to control the SLA fulfillment. However, monitoring has been signaled as a corner stone from the raising of the first SLA specifications to the more recent ones, such as WS–Agreement. Many research efforts have been made to provide information about the service level fulfillment of SLAs in order to adopt them in B2B or B2C scenarios.

Problem Statement: Once the agreement has been established between the SBSs consumers and providers, techniques to assure the agreed service quality must be developed. Such techniques require both: monitoring platforms to achieve information while the service is being consumed, and analysis engines to reason about the monitored information in order to extract useful information. Such service level fulfillment information must be clearly exposed to the parties to help in the problem solutions.

Relevance of Problem and Progress from State of the Art: The aforementioned problem has been partially covered by other works. Several proposals can be found providing *violation detection for SLAs described in WS–Agreement but in the SOA testing context, and not at service consuming time. Other proposals provide asynchronous violation detection reports to subscribed clients but they deal with ad-hoc SLA specifications. There are also proposals that go further and when they detect a SLA violation, they dynamically adapt the SBSs following different strategies. Finally, there are proposals that are able to detect and explain violation causes but with the following drawbacks: (1) the analysis of the SLA fulfillment is not performed just when a violation takes place; (2) the client (end-user) must be an expert in the reasoning paradigm to specify the SLA, but also to understand the violation explanations; (3) the violation cause is difficult to grasp by the client (application or end-user), especially when several service properties are related in the violated SLA term.*

Relation to WP Challenges: *This paper addresses two key objectives of the workpackage. The proposed approach addresses the end-to-end quality assurance at run-time, as well as the SLA conformance. To this aim, we have introduced an automated framework for monitoring and analysing SLAs.*

Solution / Research Method: *We propose SALMonADA, a service-based system (SBS) with a decoupled component architecture that integrates monitoring (SALMon) and analysis (ADA) components supporting: (1) asynchronous service monitoring of WS–Agreement documents to analyse the SLA fulfillment and report a violation when the event that causes it has been monitored (reducing the client notification time), (2) violation analysis to detect and explain the violation causes, and finally (3) notifying the clients in their own easy-to-understand specification terms. Other technical contributions are: decoupling of the monitored service technology by using an enterprise service bus (ESB); the scalability control by adding more ESBs when needed; and the use of a monitoring management document to store the monitoring information. Such a document is updated when new monitoring information is achieved.*

Benefits and Evaluation: *Those proposals, which assume the availability of a monitoring and analysis engine, benefit from using SALMonADA since they are provided with such a service level fulfillment information needed to perform various important activities, such as SBS adaptation, SLA renegotiation, and reputation statistics derivation. For demonstration purposes, we have implemented a web application as a SALMonADA client in order to specify or upload the WS-Agreement documents to monitor, execute SALMonADA and receive the results. In this web application, we have introduced the WS-Agreements of ADA and SALMon. By monitoring the SLAs of these services, we are able to assess both the functionality of SALMonADA and the non-functional aspects of its main components. Moreover, as part of the demonstration, we have simulated the consumers of the service of ADA and SALMon to prove how SALMonADA monitors, analyses and reports the service level fulfillment to their clients.*

Relation to Research Framework: *The JRA-1.2 is also closely related with our work since the SBS adaptation is one application of the service level fulfillment analysis reported by our proposal.*

Discussion and Future Work: *The conceptual model of our proposed monitoring and analysis framework is included in the paper, showing one of its instantiation by a decoupled architecture model supporting any monitoring and/or analysis component inside. We have developed such an architecture model with monitoring and analysis capabilities of previous authors proposals SALMon and ADA, respectively. Moreover, a web application client for the framework has been implemented to demonstrate the asynchronous capabilities of our proposal to monitor, analyse, and ultimately notify the service level fulfillment to the client. However, some aspects are out of the scope of this work such as the use of some temporal analysis capabilities of ADA that are currently not considered in our SALMonADA framework.*

3.2.6 "Preventing Performance Violations of Service Compositions using Assumption-based Run-time Verification" [47]

Context and Background: *Service-orientation is increasingly adopted as a paradigm to build highly dynamic, distributed applications from individual software entities, offered as services. In this work we refer to such applications as Service-based Applications, or SBAs for short. There is a clear trend that future SBAs will be increasingly composed from third-party services that are accessible over the Internet. As a consequence, SBAs will increasingly depend on the functionality and quality offered by those third parties. To prevent menacing requirements violations, SBAs should be equipped with monitoring, prediction and adaptation capabilities which are able to foresee and avert menacing violations.*

Problem Statement: *In an emergency situation, reactive adaptation can lead to critical situations, as it might delay the timely dispatch of operational forces, for example fire engines or ambulances. To address these problems, researchers have proposed to employ preventive adaptation, which enables SBAs to predict future failures and perform preventive actions. Although several approaches for preventive adaptation have been presented in the literature, they pose certain limitations, such as the need for cost models or comprehensive training data.*

Relevance of Problem and Progress from State of the Art: *Researchers have proposed to employ preventive adaptation, which enables SBAs to predict future failures and perform preventive actions. Although several approaches for preventive adaptation have been presented in the literature, they pose certain limitations, such a the need for cost models or comprehensive training data. This work aims at addressing these limitations. For critical application domains (such as emergency or financial) and important customers (such as key accounts), the SBA developer needs to ensure that each individual SBA instance will live up to its expected requirements even though its constituent, third-party services might fail.*

Relation to WP Challenges: *"Run-time Quality Assurance Techniques" and "Online Quality Prediction Techniques to Support Proactive Adaptation", as the prediction occurs during run-time.*

Solution / Research Method: *SPADE equips SBAs with adaptation capabilities, empowering them to adapt themselves preventively. To achieve this, SPADE uses run-time verification techniques, execution data of the monitored instances and assumptions concerning the SBAs' context, derived from Service Level Agreements (SLAs) of third-party services. Together, these mechanisms are used for performance prediction, which is able to detect menacing performance requirements violations of running SBAs. SPADE can thus be used in settings where no cost models or training data are available.*

Benefits and Evaluation: *Our measurement of SPADE's efficiency is twofold. First, we examined unnecessary adaptations, i.e., false positives, such as adaptations that might lead to avoidable costs, e.g., when replacing a free service with a commercial service to compensate for faults. Secondly, we count the amount of situations in which SPADE cannot perform an adaptation. It can happen that a service invocation leads to a violation of an SLA, such that the end-to-end requirement is already violated. In those situations, the SBA instance obviously cannot be adapted preventively in order to avert this requirement violation, as the requirement has already been violated. Both values are expected to be low, as a low value implies a high number of cases where SPADE was successfully applied. The conducted experiments show that SPADE actually has a small amount of false positives. Also SPADE run in just a few situations where adaptations were not possible.*

Relation to Research Framework: *The approach focuses on the Service Composition and Coordination layer (JRA-2.2), as individual services are monitored and runtime checks are performed to determine violations of e-2-e requirement. In addition, the approach is relevant to Monitoring and Adaptation (JRA-1.2), as it exploits monitoring data together with assumptions for predicting e-2-e requirement violations and performing preventive adaptation if needed.*

Discussion and Future Work: *We plan to continue our work on preventive adaptation in two directions. First, we will combine SPADE with our PROSA approach. The PROSA approach is capable of predicting quality violations of individual services. The combined approach is expected to act in situations in which SPADE is not able to prevent requirements violations as intended. Secondly, we plan to apply SPADE in a cross-layer adaptation setting. In this setting, SPADE is expected to exploit the adaptation mechanisms of two different layers: the service composition and the service infrastructure layer. We expect that harmonizing the adaptation on both layers will increase the number of situations in which SPADE is able to compensate for deviations, which thus may increase SPADE's success in avoiding requirements violations.*

3.2.7 "Future Internet Apps: The Next Wave of Adaptive Service-Oriented Systems?" [29]

Context and Background: *The Future Internet will emerge through the convergence of software services, things, content, and communication networks. Service orientation is expected to play a key role as an enabling technology that allows the provisioning of hardware and software entities and contents as services. The dynamic composition of such services will enable the creation of service-oriented systems in the Future Internet (from now on such systems are called FI Apps), which will be increasingly provided by third parties. Together with increased expectations from end-users for personalization and customization, FI Apps will thus face an unprecedented level of change and dynamism.*

Problem Statement: *The capabilities and features of FI Apps will be increasingly provided and "owned" by third parties. Examples include Internet-based software services, public sensor networks, and cloud infrastructures. Due to this "shared ownership", FI Apps will face an unprecedented level of change and dynamism. Further, expectations from end-users for what concerns the personalization and customization of those FI Apps are expected to become increasingly relevant for market success. For instance, a FI App should be able to adapt depending on the usage setting (e.g., office vs. home) or based on the available communication infrastructure (e.g., sensors vs. WiMAX). It will thus become increasingly important to engineer FI Apps in such a way that those applications can dynamically and autonomously respond to changes in the provisioning of services, availability of things and contents, as well as changes of network connectivity, end-user devices, and user expectations. Ultimately, this means that* adaptation *will become a key capability of FI Apps. Nevertheless, the characterization of the adaptation capabilities to be considered in FIApps remains a topic not fully explored.*

Relevance of Problem and Progress from State of the Art: *There has been significant progress for what concerns principles and techniques for building adaptive service-oriented systems. However, if we consider the Future Internet setting, those solutions will need to be significantly augmented, improved and integrated with a complete systems perspective. Specifically, this requires significant progress towards novel strategies and techniques for adaptation, addressing key characteristics of adaptive FI Apps. To enable the next wave of adaptive service-oriented systems in the Future Internet, it will thus be critical to understand the importance of the various adaptation characteristics to specifically target research and development activities.*

Relation to WP Challenges: *This work is relevant to the research challenge: "Quality Prediction Techniques to Support Proactive Adaptation", as it discusses how proactive adaptation triggered by predic-*

tion is expected to gain increased importance for FI Apps.

Solution / Research Method: *This work identifies and analyses key characteristics of adaptive FI Apps. Those characteristics are illustrated with examples from a scenario of the application domain of transport and logistics. The relevance of those different characteristics is scrutinized through an empirical study. As a research method we have employed an exploratory survey study, involving 51 respondents from the Future Internet community.*

Benefits and Evaluation: *We believe that the results of this survey study can enable understanding where adaptation can play a key role for FI Apps. The survey has confirmed some of the typical expectations (e.g., the importance of adaptation for service-oriented systems). However, the survey also lead to unexpected outcomes. In particular, distributed and cross-area adaptation capabilities have been deemed least important, although one would have expected those characteristics to become highly relevant in the Future Internet. We believe this deserves further investigation.*

Relation to Research Framework: *The discussion promoted in this work are related to the current elements of the S-Cube Research Framework and also with follow up issues in the scenario of Future Internet Application that consider the integration across JRA-1 and JRA-2: Monitoring and Adaptation, Engineering and Design, BPM, Service Composition and Coordination, Service Infrastructure.*

Discussion and Future Work: *One direction of future work is to better analyze the need for cross-cutting and decentralized characteristics on adaptive FI applications. We believe that there are at least two reasons to explain why these characteristics were deemed less important. First, FI areas are not consolidated enough and, currently, to the best of our knowledge, there are not proposals addressing the cross-area adaptation aspects in Web services and service-based applications, because so far it was not possible to put together IoS and IoT, for instance. But, as discussed in the beginning of this chapter, both cross-layer and cross-area are aspects related to cross-cutting adaptation characteristics in FI applications which are much deeper than the cross-layer ones in the service-oriented architecture. Second, decentralization and adaptation in the scope of Web services and service-based applications is currently associated with the very limited and self-owned environment. We believe that differently from the scope of Web services and service-based applications, FI applications assume a larger and more heterogeneous scale (e.g., Transport & Logistics applications have to deal with partners around the globe and interact with real-world devices - sensors, trucks). In this case, decentralized solutions for adaptive Web services and service-based applications proposed so far will not be suitable anymore.*

3.2.8 "Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology" [26]

Context and Background: *Adaptive capabilities are essential to guarantee the proper execution of Web services and service-oriented applications once dynamic changes are not exceptions but the rule. In fact, the importance of adaptive services significantly increases in the context of Future Internet (FI) applications. Applications in this context will have to autonomously adapt to changes on service provisioning, availability of things and content, computing resources, and network connectivity. The unprecedented level of heterogeneity and dynamic changes of FI applications will demand a transition from adaptive Web Services and service-oriented systems to adaptive FI applications.*

Three major pillars constitute the FI: the Internet of Services (IoS, e.g., software services based on third-party services), the Internet of Things (IoT, e.g., smart sensors and devices) and the Internet of Content (IoC, e.g., video streams and online games). These pillars will all converge into an integrated environment. It is expected that – to a large extent – FI applications will thus be composed of third-party offerings deployed on federated service delivery platforms through different cloud delivery models, such *as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Ultimately, this means that loosely-coupled Internet services will form a comprehensive base for developing value-added applications in an agile way. This is unlike traditional application development, which uses computing resources and software components under local administrative control. To maintain their quality of service, FI applications therefore need to dynamically and autonomously adapt to an unprecedented level of changes that may occur during runtime.*

Problem Statement: *Over the past decade a wealth of technologies for engineering adaptive Web-based services and service-oriented systems has emerged. Those technologies offer significant advancements for what concerns furnishing applications with self-adaptive capabilities. Still, those solutions focus on isolated pillars of the Future Internet only. For example, many solutions consider software services (i.e., IoS) but fall short of integrating things (i.e., IoT) which leads to different levels of heterogeneity and unprecedented level of change on available resources and data.*

Relevance of Problem and Progress from State of the Art: *Different solutions for dynamic adaptation have been developed for various areas of the FI, such as software services (IoS), as well as data and media (IoC). This work presents those solutions and suggests areas for future research to augment and integrate them towards solutions for self-adaptive FI applications.*

Relation to WP Challenges: *The contribution to the WP is related to the research challenges of "Quality Prediction Techniques to Support Proactive Adaptation" and 'Run-time Quality Assurance Techniques", as the paper discusses challenges and requirements for enabling truly adaptive FI applications. These include a seamless and consistent way of monitoring, detecting and predicting critical events.*

Solution / Research Method: *In this work, we review trends and current solutions for adaptive Web services and service-oriented applications. Based on real-world use cases from multimedia applications, as well as transport & logistics, we examine the transition from adaptive Web services to technology supporting the engineering and operation of adaptive FI applications. We demonstrate that FI applications promise full integration and combination of real, physical world services, business objectives and ICT services. This means a shift from considering adaptation only in the ICT level (and business level, as some initiatives already show), i.e., such as service-oriented applications, but extending this concept to unprecedented levels.*

Benefits and Evaluation: *This work analyzes and justifies the need for the transition from adaptive Web services and service-based applications to adaptive FI applications. We examine how current adaptive solutions need to be enhanced to properly address the adaptive needs of FI applications. Finally, we propose future challenges that need to be considered in adaptive FI applications.*

The discussions in this work are based on two real-world use cases from the multimedia and transport & Logistics domains. The first use case explores how adaptation of IoC and IoS should be considered in FI multimedia applications. The second use case demonstrates the importance of combining IoT, IoS, and business objectives for the success of FI transport & logistics applications.

Relation to Research Framework: *The contribution is relevant to Monitoring and Adaptation (JRA-1.2) and to the Service Composition and Coordination layer (JRA-2.2) as it discusses how existing solutions for monitoring services and adapting service-based applications need to be extended/integrated to meet the requirements of FI Apps.*

Discussion and Future Work: *There are many questions to be answered, and for each question new ones emerge. Despite all the uncertainties surrounding FI applications, there is at least one certain and incontestable fact: FI applications will have to be engineered explicitly considering adaptation aspects. This work discusses the need of thinking and designing FI applications considering aspects beyond the ones considered by current adaptive Web services and service-based applications. To enable adaptive FI applications, it is clear that a seamless and consistent way of monitoring, detecting and predicting critical events, dealing with cross-cutting aspects, decentralization, and boundaries between application logic and adaptation needs is required. Besides other challenges, applications need to be able to dynamically adapt to an unprecedented level of changes that can occur during runtime.*

3.2.9 "SLAs for Cross-layer Adaptation and Monitoring of Service-Based Applications: A Case Study" [5]

Context and Background: *One of the main barriers to the adoption of service-based applications (SBA) is the concern raised over the trust-worthiness and reliability of third-party services utilised in an SBA. The third-party software services are often implemented as Web services that realise business activities, such as paying with a credit card or shipping purchased goods, and they are beyond the control of the SBA provider. The problem of reliability becomes more complex when third-party cloud computing services are utilised as the underlying infrastructure for provisioning the SBA. Given that the SBA provider does not have control over the quality of the third-party services, unreliable third-party services could threaten the quality of the SBA and result in lower business performance, software faults, and performance degradation that could consequently lead to the total collapse of the SBA. Therefore the dependability of the third-party business, software, and infrastructure services utilised in an SBA becomes a principal concern for the SBA provider, who will require to adopt mechanisms within the SBA for quality assurance during run-time.*

The functional layers of an SBA have been introduced in [17] and comprise the business process management (BPM), service composition and coordination (SCC), and the service infrastructure layers (SI). As such, quality assurance approaches need to consider the layered nature of an SBA. Such an approach to the run-time quality assurance of SBAs is the cross-layer or multi-layer adaptation and monitoring (CLAM), which aims at timely detecting problems in the SBA layers and co-ordinating effective corrective actions across the SBA layers, such that problems are compensated for, or even prevented from occurring [33].

Problem Statement: *An important aspect of cross-layer adaptation and monitoring is the identification and the definition of the appropriate Service-Level Agreements (SLAs) for the third-party services utilised in the different layers of the SBAs. As such, it is necessary to analyse the SBA in order to identify the business, software, and infrastructure services and their characteristics, such that Service-Level Agreements (SLAs) are established for the third-party services. An important research question, therefore, is which process must be followed to identify the different types of third-party services and their characteristics, in order to define the appropriate SLAs in each layer.*

Relevance of Problem and Progress from State of the Art: *Recent research into run-time quality assurance has focused on implementing CLAM techniques for SBAs [33], by integrating the existing fragmented work in the field of adaptation and monitoring of service-based systems. Gjørven et al. [10] introduce a middleware for supporting the implementation of cross-layer self-adaptation of SBAs. Kazhamiakin et al. [17] describe a conceptual framework comprising the definition of the SBA layers and a set of requirements needed to be addressed by the mechanisms and techniques for CLAM of SBAs. Popescu et al. [36] present a methodology for cross-layer adaptation using adaptation templates. Latest research has focused on SLAs for CLAM. More particularly, Fugini et al. [4] describe an SLA contract that comprises parameters from user goals, business service and IT infrastructure for CLAM of*

SBAs. Schmieders et al. [48] propose the combination of SLA prediction, which uses assumptions about the characteristics of the execution context, and cross-layer adaptation mechanisms for preventing SLA violations.

SLAs for third-party services utilised in each SBA layer are an important element in such approaches, since SLAs specify the expected characteristics of each third-party service, named Service-Level Objectives (SLOs), to be monitored, and which are mapped to adaptation strategies for compensating or even proactively preventing violations of SLOs. This paper supports the research directions towards the run-time quality assurance of SBAs using CLAM techniques, while suggesting that such techniques could greatly benefit from an analysis approach of SBAs for identifying third-party services and their characteristics across the SBA layers for the definition SLAs. To the best of the authors' knowledge, there exists only one recent work that is related to the definition of SLAs for cross-layer adaptation and monitoring [8]. Although this work describes a methodology for creating SLAs, it focuses on the dependencies between the characteristics of services, and it does not follow the SBA layers as they have been defined in [33, 17]. The authors focus more on how KPIs and IT infrastructure metrics impact the goals of a service user, and they introduce a new indicator named Key Goal Indicator. In contrast, the goal of the presented paper is different since it presents ideas for defining SLAs, by performing analysis of an SBA to identify the third-party services utilised in each SBA layer and their characteristics, in order to define the appropriate SLAs in each layer.

Relation to WP Challenges: *This contribution targets the research challenges associated with the quality definition in an SBA. More specifically, it investigates the challenge of End-to-End Quality Reference Model. This contribution exemplifies an approach for establishing SLAs across the layers of an SBA, based on the characteristics of the different types of third-party services consumed in an SBA.*

Finally, this contribution is associated to a lesser extend with the challenge of Automatic quality contract establishment. Even if this contribution does not present an automated approach to contract establishment, it exemplifies the required manual steps, which could be potentially automated, for establishing SLAs across the layers of an SBA.

Solution / Research Method: *The paper present insights into how to define SLAs for CLAM, by analysing SBAs to identify the third-party business, software and infrastructure services utilised by the SBA. This paper views each layer from the perspective of the type of services utilised in the layer and it suggests that each layer concerns different types of services. The BPM layer concerns business services or business activities realised through software-based services. For instance, a shipping provider exposes a Web services API for shipping goods. The shipment of goods is a business activity provided through a Web service. The SCC layer concerns software services that implement a specific functionality or a business activity. For instance, in the case of the shipping provider, the Web service API is the software-based service. The SI layer concerns the infrastructure services used by an SBA. For instance, an SBA could be running on a third-party cloud computing infrastructure and rely on shared computing, storage, and networking resources. Based on the aforementioned suggestions, the authors argue that an SBA is a software application that outsources business activities, consumes software services, and uses infrastructure services.*

Benefits and Evaluation: *The analysis process is exemplified through a case study that concerns the definition of SLAs in an existing platform-as-a-service framework, developed during the European project CAST [18]. The analysis reveals the different third-party services and their characteristics, as a precursor to defining SLAs. In the BPM layer, two business services were identified, so two separate SLAs are required between the platform provider and the two service providers. In the SCC layer, two software services were identified, so two separate SLAs are required between the platform provider and the two service providers. In the SI layer, one infrastructure service was identified, so one SLA is required between the platform provider and the service provider.*

Each of the services identified in the study was then analysed to reveal its individual characteristics, prior to drawing up appropriate SLAs. The study clearly demonstrates the utility of separating the run-time quality assurance concerns at each layer of the SBA. The case study successfully demonstrates how distinct SLAs for business, software and infrastructure services may be applied respectively in the BPM, SCC and SI layers of an SBA, to provide the basis for building monitoring and adaptation mechanisms, which utilise SLAs across layers.

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

Discussion and Future Work: *Although the approach was exemplified through a case study, its applicability needs to be examined in more scenarios involving diverse SBAs. Additionally, the approach does not consider the existence of potential dependencies across SLAs.*

As future work the authors suggest to investigate existing methods for representing SLAs for the business, software, and infrastructure services. Finally, they plan to extend previous work [4] related to the implementation of an extensible monitoring architecture for Web services, in order to support the development of a CLAM framework for SBAs that will utilise multiple SLAs for monitoring of business, software, and infrastructure services.

3.2.10 "Negotiation towards Service Level Agreements: A Life Cycle Based Approach" [13]

Context and Background: *Service Level Agreements (SLAs) play a major role in ensuring the quality of Service Service Based System (SBSs). They stipulate the availability, reliability, and quality levels required for an effective interaction between service providers and consumers. It has been noticed that because of having conflicting priorities and concerns, conflicts arise between service providers and service consumers while negotiating over the functionality of potential services.*

Problem Statement: *Negotiation is carried out between the service provider and the consumer before any kind of agreements can be established. This negotiation is likely to raise conflicts because of difference in Quality of Service (QoS) priorities. Opposing concerns of stakeholders on the provider as well as on the consumer side, across different phases of the life cycle, may raise conflicts on negotiating QoS capabilities (such as response time). It is really important to mitigate these conflicts so that SBS stakeholders, who contribute towards the business value, can mutually agree upon an SLA. In this research, we propose a stakeholder negotiation strategy for Service Level Agreements, which is based on prioritizing stakeholder concerns based on their frequency at each phase of the SBS development life cycle.*

Relevance of Problem and Progress from State of the Art: *Conflicts which arise during the SLA negotiation are likely to be overcome either by going for an alternative service provider, or by renegotiation among stakeholders. The former one may not be a good idea as it could involve more overheads in terms of looking for a new provider and finalizing the agreement with it. Recent research in the area has not focused on stakeholders at different phases of the life cycle and their potential role in the negotiation process which eventually leads to SLAs. As they are the most common mechanism used to establish agreements on the quality of service between the service provider and the service consumer. In addition, it is important to take the stakeholders into account considering that SBSs are developed, owned, and used by different stakeholders with different perspectives, i.e. developer and provider, broker and composer, and consumer and end user respectively.*

Relation to WP Challenges: *The paper aims to address the core issue of end to end quality provision by means of SLA negotiation, which is one the focus areas of the deliverable JRA-1.3.6.*

Solution / Research Method: *We identify stakeholders and roles associated with them based on their key responsibilities at each phase of the SBS development life cycle. Then, we identify how conflicts may occur between the service provider and the consumer while negotiating towards SLA. An example scenario is presented using the Collaxa BPEL Loan Flow Service to demonstrate the potential conflicts which can occur between the stakeholders. Our goal has been twofold: propose a life cycle based negotiation methodology which could involve stakeholders, information at each phase of the SBS development life cycle, and validation of the approach in terms of numbers to demonstrate the usefulness of the approach.*

Benefits and Evaluation: *We have proposed a life cycle based methodology for negotiation between service providers and service consumers. The results suggest that assigning priority based on the proposed approach could reduce cost of SLA negotiation. We used a Loan Flow service example to identify a scenario which may lead to conflicts on QoS between service provider and the service consumer. The identified conflicting node was mapped back to the SBS life cycle to investigate the potential involvement of each life cycle phases. Using this information, we measured the preference of a stakeholders role on the conflicting node by calculating its relative frequency in comparison to the other roles. The greater the relative frequency value is, the more importance that stakeholder category has in the SLA negotiation process. These priority values were simulated to observe the potential impact of the corresponding stakeholders on the cost of the SLA negotiation.*

Relation to Research Framework: *The work presented in this paper mainly covers the theme Quality Definition, Negotiation, and Assurance which is likely to facilitate the topics of interests in JRA-2 as well; for example, Monitoring and Adaptation, Engineering and Design, Service Composition and Coordination.*

Discussion and Future Work: *We have proposed a life cycle based methodology for negotiation between service providers and service consumers. It is important to understand and implement a good negotiation process as it leads towards a formal agreement between the two parties in the form of the SLA. The numbers associated with SBS stakeholders across different like cycle phases may vary in different environments but basic theme for assigning priority to any of the stakeholder involved in the negotiation process remains the same. It should be noted that we did not include service consumers as application users as it is rather impossible to predict the exact number of the potential service users. In our future work, we plan to implement the proposed approach with automated contract negotiation. This will allow us to measure cost as well as other quality attributes associated with electronic contract negotiation. This will allow us to measure cost as well as other quality attributes associated with electronic contract negotiation*

3.2.11 "A Context-Aware Framework for Business Process Evolution" [6]

Context and Background: *Run-time adaptability of service-based business processes is a key feature of dynamic business environments, where the processes need to be constantly refined and restructured to deal with exceptional situations and changing requirements . The execution of such a system results in a set of adapted process variants instantiated on the same process model but dynamically restructured to handle specific contexts. We propose a framework supporting context-aware evolution of business processes based on process instance execution and adaptation history. Instead of looking for recurring adaptations, we propose to look for recurring adaptation needs (i.e., process instances with the same context constraint violation and system configuration). Based on the analysis of adapted instances, we* *automatically construct and rank corrective evolution variants which can handle the problematic context. At the same time, we try to identify preventive evolution variants by constructing process variants which can prevent the rasing of the adaptation need. We demonstrate the benefits of our approach using a car logistics scenario.*

Problem Statement: *The need for continuous adaptation results in a system characterized by a huge set of process executions that, although instantiated on the same process model, strongly differ in terms of process structure. Providing support for process model evolution is becoming one of the main requirements for managing the lifecycle of dynamic processes [51]. In particular, the set of adapted process instances together with the information concerning their execution should be used as training cases for evolution mechanisms in order to progressively improve process models that are then used to instantiate future process instances. Most existing approaches addressing this problem (e.g., [40, 38]) derive model-level changes by analyzing frequently occurring changes at the instance-level. In other words, if an instance-level change/adaptation occurs more frequently then a predefined threshold, the change will be propagated at the model-level. These evolution approaches present two major drawbacks. First, an instance-level adaptation variant is not good in general, as it is good for just a specific context/situation, and thus cannot simply be propagated to the process model without taking into account the adaptation need it was devised for. Second, plugging-in adaptation variants in the original process model is not always a good solution, since it may result in embedding fault-handling activities rather than trying to solve the problem that required runtime adaptation.*

Relevance of Problem and Progress from State of the Art: *The problem of supporting the evolution of business processes models has been addressed by several works. Some of these approaches are able to capture a precise sets of contexts and to use for each of them a predefined process variants [12, 41], others support the evolution of processes by analysing previous executions and adaptations [40, 38, 35, 11, 22]. Most existing approaches focus on the problem of extracting useful information from the adaptation logs. The approaches differ both in what they log and in the techniques that they use to analyze the logs (e.g., [40, 38, 11, 22, 49]). These approaches may be used in the analysis phase of the proposed evolution framework. One direction is to use process mining techniques, as in [38], considering large collections of structurally different process variants created from the same process model. The authors use a heuristic search to find a new process model such that the weighted average distance between the new model and the variants is minimal. The approach in [11] also uses mining techniques to analyze change logs. The evolution result of this approach is an abstract change process consisting of change operations and causal relations between them. These change processes can be used as an analysis tool to understand when and why changes were necessary. In [40], concepts and methods from case-based reasoning (CBR) are used in order to log, together with the change operations, also the reasons for and context of each change. Change information is stored as cases in a case-base specific to the process model. The case-bases are used to support process actors in reusing information about similar ad-hoc changes, and are also continuously monitored to automatically derive suggestions for process model changes. Change analysis and reuse is also relevant for loosely specified process models. [22] facilitates change reuse by providing a search interface for the repository of process variants. For declarative processes, [49] supports users through recommendations, which are generated based on similar past executions and considering certain optimization goals. The approaches in [40, 38] are closest to our work, since they also generate new process models based on the information from the adaptation logs. A limitation of the approach in [38] is that it does not consider the context requiring adaptation. An instance-level adaptation may be useful only for a particular context, and therefore should not be included in the process model even if it occurs relatively often. Although [40] considers also the context of changes, this is specified as natural language question-answer facts. Such facts are useful for supporting process actors in identifying the existing cases with the same context. They are also useful for the process engineer, who can manually determine the context for a new process model change.*

However, these tasks cannot be done automatically. Further, in both [40, 38] the decision whether to integrate a change operation in the process model is determined by the frequency of the change operation. However, it can be the case that the need for adaptation was a fault in the process, and the adaptation contains fault handling and compensating activities. In these cases, rather than including the adaptation in the process model, we may be interested in avoiding the adaptation need altogether. In contrast to [40, 38], we use the context as the main driver for evolution. This allows us to determine if an instance-level adaptation is useful for a particular context, and it allows us also to search for an alternative process model which avoids a problematic context. The importance of context for improving process models is recognized also in [35]. Here, the authors propose a context-aware process management cycle, with context-awareness spanning all the stages of the process lifecycle. While [35] remains at a very general and abstract level, we are taking the approach one step further, and provide concrete ideas for implementing the context-aware process management cycle.

Relation to WP Challenges: *The framework proposed in the paper contributes to the challenge "Quality Prediction Techniques to Support Proactive Adaptation". This because its main goal is to evolve service-based business process (i.e., long-term adaptation) considering the set of adapted process instances together with the information concerning their success and their execution context. All this collected information are used as training cases for a performance analysis in order to progressively improve the process models with respect to a set of KPIs. The results of this analysis can be used to generate a set of evolution variants which optimize the existing process model with respect to the KPIs. The evolution variants are then proposed to the process designer. In case the process designer decides to evolve the process model, all new process instances will be based on the evolved model.*

Solution / Research Method: *To overcome the limitations, presented in the problem statement paragraph, we present a context-aware evolution framework that, instead of searching for recurring process changes, searches for recurring adaptation needs. At run-time, we may determine that a certain context which at design-time was assumed to occur rarely, actually occurs for a high percentage of the process' instances. In this situation, we analyze the instance-level adaptations that have been used for handling the unexpected context. Based on these adaptations, we determine the changes that must be performed on the process model, in order to handle the new context. On the basis of the analysis results, the framework automatically proposes process variants that either embed general corrective solutions derived from context-specific adaptations, or restructure the original model to prevent the recurring adaptation need from occurring. Figure 3.1 shows an architectural overview of the framework presenting the relations between the different components and positioning them with respect to the the three main phases of the evolution lifecycle, namely (1)* execution phase*, (2)* analysis phase*, and (3)* evolution phase.

During the first phase, the framework is responsible for managing the execution and adaptation of the system and for logging all the information that may be useful to the other phases (e.g., execution traces, adaptation needs, adaptation variants, execution performances). During the analysis phase, the framework controls and evaluates the quality of execution of the processes with respect to the KPIs, decides the need for evolution for a certain process model, and, on the basis of the execution history, identifies the contextual evolution problem in terms of recurring system configuration that required adaptation. Finally, in the evolution phase, the framework uses the information obtained from the analysis phase to compute process model variants that either embed the best performing (with respect to KPIs) adaptation variants or prevent the violation of the context constraint. These evolution variants are then presented to the process designer, who decides whether they should be adopted for future executions. The process designer obtains the evolved process model using a set of supporting tools.

Benefits and Evaluation: *We have presented a framework for evolving process models based on a history of process instance executions and adap- tations. Our approach is context-driven. If the need to evolve the process model is detected, we analyze the relevant adapted process instances and look for*

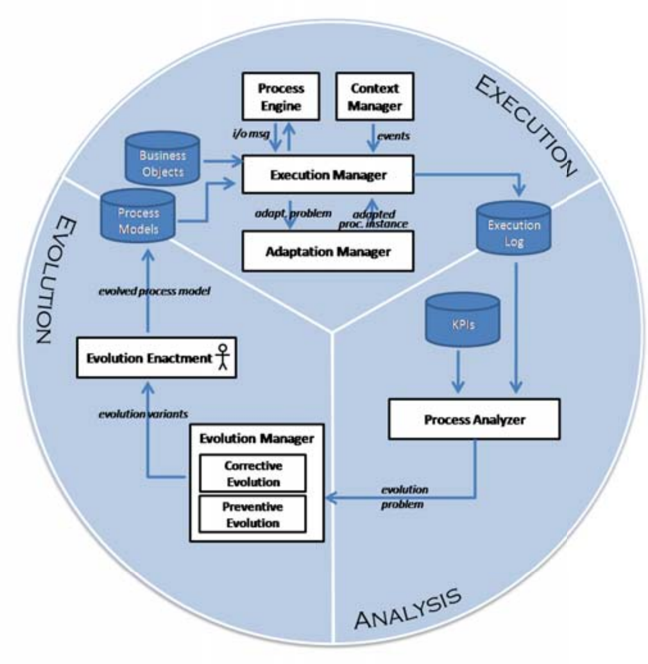

Figure 3.1: Evolution Framework

recurring adaptation needs (i.e., the same constraint violation and system configuration). This allows us to construct and rank evolution variants which can handle the problematic context (corrective evolution). It also allows us to construct evolution variants which can prevent the adaptation need (preventive evolution). We introduced the modeling artifacts for evolvable process-based applications, and a detailed framework for context- aware evolution. Finally, we demonstrate the benefits of our approach using a car logistics scenario.

Relation to Research Framework: *The proposed approach is related with both WP-JRA 1.1 and WP-JRA 1.2. With respect WP-JRA 1.1, our approach covers both the adaptation and evolution part of the S-Cube life-cycle to manage adaptable SBA. This life-cycle was refined with explicit artifacts needed to specify and manage adaptable and evolvable service-based business processes. Concerning WP-JRA 1.2 we re-use some results already presented in the WP concerning context-aware adaptation of service-based business processes.*

Discussion and Future Work: *We have presented a framework for evolving process models based on a history of process instance executions and adaptations. Our approach is context-driven. If the need to evolve the process model is detected, we analyze the relevant adapted process instances and look for recurring adaptation needs (i.e., the same constraint violation and system configuration). This allows us to construct and rank evolution variants which can handle the problematic context (corrective evolution). It also allows us to construct evolution variants which can prevent the adaptation need (preventive evolution). In our future work, we will develop concrete solutions for corrective and preventive evolution. For corrective evolution we plan to develop techniques to automatically transform instance-level adaptation variants into evolution variants using the built-in adaptation tools. For preventive evolution we will apply AI planning techniques to re-plan the process model in order to avoid the critical configurations. We will implement and evaluate our solutions on realistic scenarios, such as the car logistic scenario introduced in this work.*

3.2.12 "Constraint-Based Runtime Prediction of SLA Violations in Service Orchestrations" [16]

Context and Background: *Quality of Service (QoS), such as execution time, availability, reliability or cost, is crucial for the usability and economic value of services and service compositions. Service Level Agreements (SLAs) define the QoS levels expected to be met by a service provider to the service clients. SLAs usually specify ranges of values of QoS properties in the form of Service Level Objectives (SLOs), which represent the constraints on the design and behavior of services and service compositions. Predicting an SLO violation ahead can trigger adaptation to avoid this violation by changing the data or behavior of an executing instance.*

Problem Statement: *The problem addressed in this contribution is efficient and accurate prediction (ahead of time) violations of SLA objectives (SLOs) for a running instance of a service orchestration. Furthermore, we are concerned with fine-grained predictions that can be made at arbitrary points in the execution, and even continually. We also want to minimize the dependence on historic data (that is usually used in the data-mining approaches) and concentrate on the structure of the orchestration and the tasks ahead in its execution.*

Relevance of Problem and Progress from State of the Art: *The proposed approach makes a detour from the usual statistical prediction models for SLA violation that typically require significant amount of data on both component services in an orchestration, as well as on the behavior of the orchestration itself. Several such approaches, based on decision tree induction or regression models have been proposed [52, 21]. While such approaches in principle abstract from the structure of a service orchestration and avoid dependencies on particular orchestration language and constructs, they have several problems that the present approach aims at solving. First, the data mining techniques used in these approaches use depend on historic probability distributions, which tend to reflect past conditions in the service system and its environment, rather than the current ones. Adding more information to the statistical prediction models is computationally expensive, and therefore updating the model is done only periodically, and usually not on a running instance level. If the definition of the orchestration changes, the collected historic data becomes obsolete, and new critical mass of data has to be collected. Furthermore, if the structure of an executing instance changes, data mining based predictions become unreliable, unless a model that corresponds to the new structure has been precomputed.*

The constraint based approach to QoS prediction also relies on historic data that describes the ranges of QoS values for component services, but in a different manner. First, it is not concerned with probability distributions inside of these ranges, or with stochastic (in)dependence between QoS of different services and their joint probability distributions that require significant data sets to be statistically significant, and encode historic trends. Second, updating the bounds from new exection data is quite a simple operation in comparison to (re)building of data mining models, and can be performed to update accuracy of predictions as soon as such data is available.

Instead of trying to historically correlate orchestration QoS and that of its components, the constraint based approach relies on the structure and the current state and control point in execution of a given instance of the orchestration. That information is always implicitly present in the interpreter (i.e., the orchestration execution engine), from which it can be extracted either by design of the engine, or in principle (e.g., by "doctoring" the existing open-source engines). Since the prediction is made at the instance level, and can be performed continually, an instance-level adaptation does not decrease the accuracy of prediction. Furthermore, the prediction is based on efficient and industry proven Constraint Logic Programming [2] tools and techniques that present little overhead on top of the regular execution.

Relation to WP Challenges: *This contribution is related to the following research challenge within the workpackage JRA-1.3:*

- *Quality Prediction Techniques to Support Proactive Adaptation*

Solution / Research Method: *The solution is based on using a continuation [39] that is emitted by the process execution engine, which describes the steps in the service orchestration that remain to be performed from the moment of observation until the end of its execution (including remaining iterations of the started loops). The continuation is used for constructing a set of equalities and inequalities (i.e., the constraints) over variables that represent QoS metrics [3] (such as running time or availability) of the building blocks in the continuation, starting from elementary state, and upwards. For service invocations, predefined upper and lower bounds for QoS are used, which can be updated by observing the actual invocation and reply events exchanged between the execution engine and the external services. For if-then-else branches, the Boolean value of the condition is explicitly included into the constraints to allow reasoning on what conditions may lead to SLO compliance and violation, respectively. For loops, an integer loop counter is explicitly introduced into the constraints to enable reasoning on what number of loop iterations may lead to SLO compliance and violation, respectivelly. The loop counters can be further constrained using either known data at run-time to compute their exact value (such as in a* `foreach` *over data structures of the known size), or by performing computational cost analysis and expressing the (safe and conservative) upper and lower bounds for loop iterations as functions of data in the orchestration [15, 14, 30].*

The set of constraints generated from the given continuation (together with the constrained variables and their domains) constitutes a Constraint Satisfaction Problem (CSP) [7, 1] that is solved, using an interval constraint solver [2], for both the case of SLO compliance and the case of SLO violation. Existence of solution just in one of these two cases signifies prediction of certain SLO compliance or violation, respectively. When solutions for both cases exist, the differences between the resulting intervals for the constrained variables (for instance, branch conditions or loop iteration counters) can be used for detecting events that necessarily lead to the one or the other outcome. For instance, the predictor can say that an SLO may be satisfied if the number of loop iterations is below 12, and that the SLO may be violated if the number of loop iterations is greater than 2. This allows us to infer, e.g., that exiting the loop after less than three iterations leads to the SLO's satisfaction, and that at entering the twelfth iteration the SLO violation is imminent. The ranges for all QoS metric variables included in the CSP, including all component metrics, are given as numeric intervals.

Benefits and Evaluation: *We have evaluated the proposed constraint-based QoS predictor on a sample industrial service composition that involves collection of purchase orders, product planning, assembly, billing and delivery. We have observed 100 instances and used the information on the execution ranges of component services to predict whether the composition's time limit (the SLA objective under consideration) can be met and under what conditions.*

The results have shown that performing around 160 predictions during the lifetime of an average instance takes between 1 and 2% of its execution time, thus proving that this prediction technique does not incur significant computational overheads. The accuracy of prediction, across different time limits, ranges between 94% and 100%, provided that the assumptions on the running times of the component services are correct. The time lead between a SLO failure prediction and the actual violation ranged on average between 15% and 20% of the time limit.

Relation to Research Framework: *Relative to the service life-cycle, the proposed constraint-based predictor of SLA violations is applied at service* composition run-time. *The prediction is applied at the* Service Composition Layer, *and relates to the* End-to-End Quality Assurance *as the cross-cutting concern in the IRF.*

Discussion and Future Work: *The constraint based prediction of SLA violations for service orchestration is a very accurate and efficient QoS prediction technique that can be applied to individual service* *instances and can accommodate both static and dynamic instance-level adaptation. It is subject to availability of process continuations, which are provided by or extracted from the process execution engine. Its accuracy is affected by the knowledge of safe QoS bounds for component services, which can start from initial assumptions and be continually updated from the new data.*

Our future work will concentrate on applying this approach to the existing process execution engines, and on evaluating the effect of incomplete and imprecise knowledge on the precision and accuracy of prediction.

Chapter 4

Conclusions

Adaptation has been long been proposed as a solution to ensure that service-based applications become resilient against failures and changes of third-party services. For those service-based applications, we see a recent trend to complement solutions for reactive adaptation (i.e., repairing a system in response to failures that have actually occurred) with proactive capabilities (i.e., modifying the system before an imminent failure actually occurs). Proactive adaptation thus means that the service-based application "can try to apply countermeasures in order to prevent the occurrence of a failure, or it can prepare repair mechanisms for the upcoming failure in order to reduce time-to-repair" [44].

This deliverable presented 12 contributions that introduce novel and improved (based on validation results) approaches for proactive negotiation and quality assurance, providing essential capabilities for proactive adaptation. Out of the 12 publications (with overlaps), 3 focus on quality negotiation and 10 focus on run-time quality assurance and quality prediction.

Clearly, the majority of the presented contributions is concerned with principles and techniques for online quality prediction in the context of 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.

4.1 Future Research Activities on Online Quality Prediction

Adaptation capabilities will become even more relevant in the extremely dynamic and complex setting envisioned in the Future Internet, where – besides others – services provided by the Internet of Services (IoS) and the Internet of Things (IoT) will converge and will thus jointly form service-based applications.

Those trends mean that opportunities and benefits for proactive adaptation will even increase. However, the Future Internet will amplify existing research issues on online quality prediction, as well as lead to additional research issues. As a final outcome of S-Cube, the network will publish a research roadmap looking in detail at those issues (see http://www.s-cube-network.eu/icse). To give a flavour, some of these issues include:

- *Ensuring that online failure prediction is accurate is critical [28]. Otherwise, wrong predictions may lead to the execution of unnecessary adaptations (false positives) or missed adaptation opportunities (false negatives). As an example, unnecessary adaptations may introduce severe problems; e.g., if a working service is replaced by one with bugs. Providing accurate failure predictions becomes extremely challenging in the setting of Future Internet applications, especially if they consist of third-party IoS and IoT services due to the heterogeneity and dynamicity of the entities.*
- *Traditionally, accuracy of predictions is assessed in a "post-mortem" may such as to select a matching prediction technique for a specific usage setting. However, in the Future Internet those usage settings or contexts will continuously change. This means that even if high accuracy is*

achieved in an initial setting, accuracy may quickly decrease over time. We thus need new ways to assess the accuracy online, such as to determine during run-time whether to trust the predictions.

- *The challenges towards online quality prediction are further amplified in the presence of noisy and uncertain data. Open issues involve how to reason and predict in the presence of such noisy data and to understand how such uncertainties may impact on the accuracy of predictions.*
- *The Future Internet will lead to a proliferation of data sources and an increased amount and timeliness of operational data (aka. "Big Data" phenomenon). On the one hand, such data may provide better prediction techniques, as more data is available to reason on future situations. On the other hand, "Big Data" leads to issues such as which of the data to store, what to filter and how to process huge data streams in real-time.*

***

Acknowledgments: *This deliverable constitutes the final document of WP-JRA-1.3. As workpackage leader, I would like to extend my gratitude to all researchers that jointly produced excellent, novel research outcomes concerning the challenges pursued within the workpackage and with whom I had many fruitful collaborations and inspiring discussions.* —Andreas Metzger

Bibliography

[1] *K. R. Apt.* Principles of Constraint Programming. *Cambridge University Press, 2003.*

[2] *Krzysztof R. Apt and Mark G. Wallace.* Constraint Logic Programming Using ECLIPSE. *Cambridge University Press, 2007.*

[3] *S. Benbernou, I. Brandic, C. Cappiello, M. Carro, M. Comuzzi, A. Kertész, K. Kritikos, M. Parkin, B. Pernici, and P. Plebani. Modeling and negotiating service quality. In* Service Research Challenges and Solutions for the Future Internet, *volume 6500 of* LNCS, *pages 157–208. Springer Verlag, 2010.*

[4] *Konstantinos Bratanis, Dimitris Dranidis, and Anthony J. H. Simons. An extensible architecture for run-time monitoring of conversational web services. In* Proceedings of the 3rd International Workshop on Monitoring, Adaptation and Beyond, *MONA '10, pages 9–16, New York, NY, USA, 2010. ACM.*

[5] *Konstantinos Bratanis, Dimitris Dranidis, and Anthony J. H. Simons. Slas for cross-layer adaptation and monitoring of service-based applications: a case study. In* Proceedings of the International Workshop on Quality Assurance for Service-Based Applications, *QASBA '11, pages 28–32, New York, NY, USA, 2011. ACM.*

[6] *A. Bucchiarone, A. Marconi, M. Pistore, and A. Sirbu. A context-aware framework for business processes evolution. In* Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2011 15th IEEE International, *pages 146 –154, 29 2011-sept. 2 2011.*

[7] *Rina Dechter.* Constraint Processing. *Morgan Kauffman Publishers, 2003.*

[8] *Mariagrazia Fugini and Hossein Siadat. SLA contract for Cross-Layer monitoring and adaptation. In Stefanie Rinderle-Ma, Shazia Sadiq, and Frank Leymann, editors,* Business Process Management Workshops, *volume 43, pages 412–423. Springer Berlin / Heidelberg, 2010.*

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

[10] *Eli Gjørven, Romain Rouvoy, and Frank Eliassen. Cross-layer self-adaptation of service-oriented architectures. In* Proceedings of the 3rd workshop on Middleware for service oriented computing, *MW4SOC '08, pages 37–42. ACM, 2008.*

[11] *Christian W. Guenther, Stefanie Rinderle-Ma, Manfred Reichert, Wil M.P. van der Aalst, and Jan Recker. Using process mining to learn from process changes in evolutionary systems.* Int'l Journal of Business Process Integration and Management, *3(1):61–78, 2007.*

[12] *Alena Hallerbach, Thomas Bauer, and Manfred Reichert. Capturing variability in business process models: the provop approach.* Journal of Software Maintenance, *22(6-7):519–546, 2010.*

[13] *Sajid Ibrahim Hashmi, Rafiqul Haque, Eric Schmieders, and Ita Richardson. Negotiation towards service level agreements: A life cycle based approach. In* SERVICES'11*, pages 1–8, 2011.*

[14] *D. Ivanović, M. Carro, and M. Hermenegildo. An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Proactive Monitoring. In* Pre-proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+)*, November 2009.*

[15] *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.*

[16] *Dragan Ivanovic, Manuel Carro, and Manuel V. Hermenegildo. Constraint-based runtime prediction of sla violations in service orchestrations. In* ICSOC'11*, pages 62–76, 2011.*

[17] *Raman Kazhamiakin, Marco Pistore, and Asli Zengin. Cross-layer adaptation and monitoring of service-based applications. In* Proceedings of the 2009 international conference on Service-oriented computing*, ICSOC/ServiceWave'09, pages 325–334, Berlin, Heidelberg, 2009. Springer-Verlag.*

[18] *D. Kourtesis, V. Kuttruff, and I. Paraskakis. Optimising development and deployment of enterprise software applications on PaaS: the CAST project. In* ServiceWave 2010 Workshop Proceedings*. Springer Berlin / Heidelberg, 2010.*

[19] *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.*

[20] *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.*

[21] *Philipp Leitner, Anton Michlmayr, Florian Rosenberg, and Schahram Dustdar. Monitoring, prediction and prevention of sla violations in composite services. In* ICWS*, pages 369–376. IEEE Computer Society, 2010.*

[22] *Ruopeng Lu and Shazia Wasim Sadiq. Managing process variants as an information resource. In* Proc. BPM'06*, pages 426–431, 2006.*

[23] *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.*

[24] *C. M*

[25] *K. Mahbub and G. Spanoudakis. Proactive sla negotiation for service based systems: Initial implementation and evaluation experience. In* Services Computing (SCC), 2011 IEEE International Conference on*, pages 16 –23, july 2011.*

[26] *C. C. Marquezan, A. Metzger, K. Pohl, V. Engen, M. Boniface, S.C. Phillips, and Z. Zlatev.* Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology*. IGI, 2012. accepted for publication.*

[27] *Andreas Metzger. Towards accurate failure prediction for the proactive adaptation of service-oriented systems (invited). In* Proceedings Workshop on Assurances for Self-Adaptive Systems (ASAS), collocated with ESEC 2011*, Szeged, Hungary, 2011.*

[28] *Andreas Metzger. Towards accurate failure prediction for the proactive adaptation of service-oriented systems (invited). In* Proceedings Workshop on Assurances for Self-Adaptive Systems (ASAS), collocated with ESEC 2011*, Szeged, Hungary, 2011.*

[29] *Andreas Metzger and Clarissa Cassales Marquezan. Future internet apps: the next wave of adaptive service-oriented systems? In* Proceedings of the 4th European conference on Towards a service-based internet*, ServiceWave'11, pages 230–241, Berlin, Heidelberg, 2011. Springer-Verlag.*

[30] *J. Navas, E. Mera, P. López-García, and M. Hermenegildo. Inference of User-Definable Resource Bounds Usage for Logic Programs and its Applications. Technical Report CLIP5/2008.0, Technical University of Madrid (UPM), School of Computer Science, UPM, July 2008.*

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

[32] *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.*

[33] *Mike Papazoglou, Klaus Pohl, Michael Parkin, and Andreas Metzger, editors.* Service research challenges and solutions for the future internet: S-cube - towards engineering, managing and adapting service-based systems*. Springer-Verlag, Berlin, Heidelberg, 2010.*

[34] *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.*

[35] *K. Ploesser, M. Peleg, P. Soffer, M.Rosemann, and J. Recker. Learning from context to improve business processes. 6(1):1–7, 2009.*

[36] *Razvan Popescu, Athanasios Staikopoulos, Peng Liu, Antonio Brogi, and Siobhan Clarke. Taxonomy-Driven adaptation of multi-layer applications using templates. In* Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 213–222, Los Alamitos, CA, USA, 2010. IEEE Computer Society.*

[37] *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.*

[38] *Manfred Reichert, Stefanie Rinderle, and Peter Dadam. On the common support of workflow type and instance changes under correctness constraints. In* CoopIS'03*, pages 407–425, 2003.*

[39] *John C. Reynolds. The discoveries of continuations.* LISP and Symbolic Computation Journal*, 6:233–247, 1993.*

[40] *Stefanie Rinderle, Barbara Weber, Manfred Reichert, and Werner Wild. Integrating process learning and process evolution - a semantics based approach. In* Proc. BPM'05*, pages 252–267, 2005.*

[41] *Marcello La Rosa, Wil M. P. van der Aalst, Marlon Dumas, and Arthur H. M. ter Hofstede. Questionnaire-based variability modeling for system configuration.* Software and System Modeling*, 8(2):251–274, 2009.*

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

[43] *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.*

[44] *Felix Salfner, Maren Lenk, and Miroslaw Malek. A survey of online failure prediction methods.* ACM Comput. Surv.*, 42:10:1–10:42, March 2010.*

[45] *O. Sammodi, A. Metzger, X. Franch, M. Oriol, J. Marco, and K. Pohl. Usage-based online testing for proactive adaptation of service-based applications. In* 35th Annual IEEE Computer Software and Applications Conference (COMPSAC)*, pages 582 –587, Los Alamitos, CA, USA, july 2011. IEEE Computer Society.*

[46] *Osama Sammodi, Andreas Metzger, Marc Oriol, Jordi Marco, Xavier Franch, and Klaus Pohl. Accurate service failure prediction through online testing. Submitted to ICWS'12, 2012.*

[47] *Eric Schmieders and Andreas Metzger. Preventing performance violations of service compositions using assumption-based run-time verification. In* Proceedings of the 4th European conference on Towards a service-based internet*, ServiceWave'11, pages 194–205, Berlin, Heidelberg, 2011. Springer-Verlag.*

[48] *Eric Schmieders, Andras Micsik, Marc Oriol, Khaled Mahbub, and Raman Kazhamiakin. Combining SLA prediction and cross layer adaptation for preventing SLA violations. In* Proceedings of the 2nd Workshop on Software Services: Cloud Computing and Applications based on Software Services*, Timisoara, Romania, June 2011.*

[49] *Helen Schonenberg, Barbara Weber, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. Supporting flexible processes through recommendations based on history. In* Proc. BPM'08*, pages 51–66, 2008.*

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

[51] *Barbara Weber, Shazia Wasim Sadiq, and Manfred Reichert. Beyond rigidity - dynamic process lifecycle support.* Computer Science - R&D*, 23(2):47–65, 2009.*

[52] *Branimir Wetzstein, Philipp Leitner, Florian Rosenberg, Ivona Brandic, Schahram Dustdar, and Frank Leymann. Monitoring and analyzing influential factors of business process performance. In* EDOC*, pages 141–150. IEEE Computer Society, 2009.*