Grant Agreement N° 215483

Title:	***Mechanisms and Techniques for QoS-Aware, Coordinated Service Compositions***
Authors:	*UoC, TUW, UniHH, UPM, USTUTT, UniDuE, FBK, UCBL-UPD*
Editor:	*Dragan Ivanović (UPM)*
Reviewers:	*Annapaola Marconi (FBK)* *Harald Psaier (TUW)* *Dimka Krastoyanova (USTUTT)*
Identifier:	*CD-JRA-2.2.6*
Type:	*Contractual Deliverable*
Version:	*1.0*
Date:	*15 February 2012*
Status:	*Final*
Class:	*External*

Management Summary

This deliverable presents the contributions to technical foundations of self-configuring, adaptive, QoS-aware service compositions. The contributions deal with monitoring and analysis of service compositions and choreographies and their QoS characteristics in the scope of cross-organisational business processes. The role of QoS characteristics of service compositions are considered in the scope of Quality Assurance and thus foster integration with other workpackages.

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

Members of the S-Cube consortium:

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

Published S-Cube documents

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

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

The S-Cube Deliverable Series

Vision and Objectives of S-Cube

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

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

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

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

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

This page intentionally left blank.

Contents

1 Deliverable Overview **8**
1.1 Introduction . . . 8
1.2 Deliverable Structure . . . 8
1.3 Overview of the Contributions . . . 9
1.4 Key Research Challenges and Results . . . 11
1.5 Relationship to Other Work Packages . . . 12

2 Mechanisms and Techniques for QoS-Aware, Coordinated Service Compositions **14**
2.1 Awareness-Based Realizability Analysis Of Choreographies With Exception Handling Constructs . . . 14
2.1.1 Background . . . 14
2.1.2 Problem Statement . . . 15
2.1.3 Contribution Relevance . . . 15
2.1.4 Contribution Summary . . . 15
2.1.5 Contribution Evaluation . . . 15
2.1.6 Relation to the Research Framework . . . 15
2.1.7 Conclusions . . . 15
2.2 Automatic Attribute Inference In Complex Orchestrations Based On Sharing Analysis . . . 16
2.2.1 Background . . . 16
2.2.2 Problem Statement . . . 16
2.2.3 Contribution Relevance . . . 16
2.2.4 Contribution Summary . . . 17
2.2.5 Contribution Evaluation . . . 17
2.2.6 Relation to the Research Framework . . . 17
2.2.7 Conclusions . . . 17
2.3 Towards Deriving Specifications for Composite Web Services . . . 18
2.3.1 Background . . . 18
2.3.2 Problem Statement . . . 18
2.3.3 Contribution Relevance . . . 18
2.3.4 Contribution Summary . . . 19
2.3.5 Contribution Evaluation . . . 19
2.3.6 Relation to the Research Framework . . . 19
2.3.7 Conclusions . . . 19
2.4 Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitutions . . . 20
2.4.1 Background . . . 20
2.4.2 Problem Statement . . . 20
2.4.3 Contribution Relevance . . . 20
2.4.4 Contribution Summary . . . 21
2.4.5 Contribution Evaluation . . . 21

2.4.6 Relation to the Research Framework . . . 21
2.4.7 Conclusions . . . 21
2.5 Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Adaptation . . . 22
2.5.1 Background . . . 22
2.5.2 Problem Statement . . . 22
2.5.3 Contribution Relevance . . . 22
2.5.4 Contribution Summary . . . 22
2.5.5 Contribution Evaluation . . . 23
2.5.6 Relation to the Research Framework . . . 23
2.5.7 Conclusions . . . 23
2.6 A Context-Aware Framework for Business Processes Evolution . . . 24
2.6.1 Background . . . 24
2.6.2 Problem Statement . . . 24
2.6.3 Contribution Relevance . . . 25
2.6.4 Contribution Summary . . . 25
2.6.5 Contribution Evaluation . . . 25
2.6.6 Relation to the Research Framework . . . 25
2.6.7 Conclusions . . . 25
2.7 Dynamic Composition of Pervasive Process Fragments . . . 26
2.7.1 Background . . . 26
2.7.2 Problem Statement . . . 26
2.7.3 Contribution Relevance . . . 26
2.7.4 Contribution Summary . . . 27
2.7.5 Contribution Evaluation . . . 27
2.7.6 Relation to the Research Framework . . . 27
2.7.7 Conclusions . . . 28
2.8 Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction . . . 28
2.8.1 Background . . . 28
2.8.2 Problem Statement . . . 28
2.8.3 Contribution Relevance . . . 29
2.8.4 Contribution Summary . . . 29
2.8.5 Contribution Evaluation . . . 29
2.8.6 Relation to the Research Framework . . . 29
2.8.7 Conclusions . . . 30
2.9 An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering . . . 30
2.9.1 Abstract . . . 30
2.9.2 Background . . . 30
2.9.3 Problem Statement . . . 30
2.9.4 Contribution Relevance . . . 31
2.9.5 Contribution Summary . . . 31
2.9.6 Contribution Evaluation . . . 31
2.9.7 Relation to the Research Framework . . . 31
2.9.8 Conclusions . . . 31
2.10 Tweetflows - Flexible Workflows with Twitter . . . 32
2.10.1 Background . . . 32
2.10.2 Problem Statement . . . 32
2.10.3 Contribution Relevance . . . 32

2.10.4 Contribution Summary . . . 32
2.10.5 Contribution Evaluation . . . 33
2.10.6 Relation to the Research Framework . . . 33
2.10.7 Conclusions . . . 33
2.11 A Penalty-based Approach for QoS Dissatisfaction using Fuzzy Rules . . . 33
2.11.1 Background . . . 34
2.11.2 Problem Statement . . . 34
2.11.3 Contribution Relevance . . . 34
2.11.4 Contribution Summary . . . 34
2.11.5 Contribution Evaluation . . . 34
2.11.6 Relation to the Research Framework . . . 35
2.11.7 Conclusions . . . 35

3 Conclusions **36**
3.1 Summary . . . 36
3.2 An Outline of Future Work . . . 37

Bibliography **38**

A Attached Papers **42**
A.1 Awareness-Based Realizability Analysis Of Choreographies With Exception Handling Constructs . . . 43
A.2 Automatic Attribute Inference In Complex Orchestrations Based On Sharing Analysis . . . 116
A.3 Towards Deriving Specifications for Composite Web Services . . . 124
A.4 Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitutions . . . 134
A.5 Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Adaptation . . . 149
A.6 A Context-Aware Framework for Business Process Evolution . . . 165
A.7 captevo: Context-aware Adaptation and Evolution of Business Processes . . . 174
A.8 Dynamic Composition of Pervasive Process Fragments . . . 176
A.9 Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction . . . 184
A.10 An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering . . . 204
A.11 Tweetflows - Flexible Workflows with Twitter . . . 219
A.12 A Penalty-based Approach for QoS Dissatisfaction using Fuzzy Rules . . . 226

Chapter 1

Deliverable Overview

1.1 Introduction

The goal of the S-Cube work package WP-JRA-2.2 is to establish the foundation for Quality-of-Service (QoS) aware adaptable service compositions. Such compositions are able to adapt themselves in response to changes in the QoS characteristics of the invoked services and systems within which they execute and with which they communicate. QoS characteristics of a services are specified by different Service-Oriented Architecture (SOA) functional layers, from the infrastructure, to individual "atomic" services, properties of other services, to high-level business processes. Therefore, service compositions and their adaptation are influenced by a combination of those factors. The constraints to which the actual QoS of an executing service instance are expected to conform are usually defined using Service Level Agreements (SLA). The work included in this work package thus relies on inputs, capabilities, and QoS requirements from both the Business Process Management and the Service Infrastructure SOA functional layers, and uses them throughout service composition lifecycle, including modeling, verification, monitoring, and adaptation.

The goal of this deliverable is to present mechanisms and techniques for QoS-aware, coordinated service compositions. That includes the technical foundations of self-configuring, adaptive, QoS-aware service compositions, and the work on monitoring and analysis of service compositions and choreographies and their QoS characteristics in the scope of cross-organisational business processes. The work investigates how such monitoring results can be used for prediction of fluctuations in QoS characteristics and the potential selection of adaptation strategies and mechanisms for the overall Service-Based Application (SBA).

Being the final deliverable in the S-Cube WP-2.2 series, its important objective is integration of the results with the other work packages within S-Cube. Especially, the role of QoS characteristics of service compositions will be investigated in the context of Quality assurance and thus foster integration with other WPs, most notably WP-JRA-1.3, *End-to-End Quality Provision and SLA Conformance*.

1.2 Deliverable Structure

This deliverable is based on scientific publications developed within the framework of S-Cube. This first introductory chapter gives an overview of the contributed publications, describes how they relate to the goal of this deliverable, positions the contributions in the S-Cube Integrated Research Framework (IRF), and puts them in relationship with other S-Cube work packages.

Chapter 2 on page 14 gives a detailed description of each contribution included in the deliverable. Besides the basic information on the contribution title, the participating S-Cube partners, keywords related to the content, and the state of submission/publication, each description provides background, problem statement, contribution relevance to the deliverable, its summary, evaluation and conclusions. The actual

papers are attached in the appendix at the end of the deliverable.

Finally, Chapter 3 on page 36 gives a brief summary of the findings contained in the contributions, and provides an outline of future work. It is followed by the bibliography.

1.3 Overview of the Contributions

This deliverable includes the following 11 contributions:

- *Awareness-Based Realizability Analysis Of Choreographies With Exception Handling Constructs*
- *Automatic Attribute Inference In Complex Orchestrations Based On Sharing Analysis*
- *Towards Deriving Specifications for Composite Web Services*
- *Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitutions*
- *Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Adaptation*
- *A Context-Aware Framework for Business Processes Evolution*
- *Dynamic Composition of Pervasive Process Fragments*
- *Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction*
- *An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering*
- *Tweetflows - Flexible Workflows with Twitter*
- *A Penalty-based Approach for QoS Dissatisfaction using Fuzzy Rules*

The contributions presented in this deliverable present a number of techniques and approaches applicable to QoS-aware, coordinated service compositions, from different perspectives and taking into account various relevant aspects of the Service-Oriented Architecture (SOA). They are motivated by scenarios and goals of service providers, users, designers, and developers that appear in different stages of SBA life-cycle. The contributions extend the work that has been published and presented in the earlier deliverables in the S-Cube JRA-2.2 series on fragmentation [7], coordinated service compositions [45], and derivation of QoS specifications for services and service compositions [12]. Relationship of the contributions presented in this deliverable is discussed in section 1.5 *Relationship to Other Work Packages* on page 12. Detailed descriptions of the contributions can be found in Chapter 2 on page 14, and the actual papers are attached in the Appendix (starting at page 42). In the remainder of this section, we give a brief overview of the key highlights.

Using and Developing Formal Methods

Several contributions rely on the existing formal methods and models (or the development of new ones) for analyzing/verifying properties of service compositions, and/or synthetizing service compositions from specifications. *Awareness-Based Realizability Analysis Of Choreographies With Exception Handling Constructs* (p. 14) addresses the issue of detecting defects in realization of service choreography specifications (built from process algebra formulas) and providing precise diagnostics that can help service designers and developers to detect statically (at design-time) when their choreography specification cannot be realized, and to facilitate the changes and evolution of these specifications towards attaining realizability.

Automatic Attribute Inference In Complex Orchestrations Based On Sharing Analysis (p. 16) uses the notion of data sharing to model behavior of individual activities, complex constructs in a service orchestration (branches, loops, sub-workflows) with respect to their input and output data. It employs static analysis to infer which (user-defined) attributes of input data to a service orchestration are (potentially) shared by individual activities and intermediate/resulting data items in the orchestration. That information can be used for, e.g., splitting the orchestration into fragments for distributed enactment on the basis of information flow.

Towards Deriving Specifications for Composite Web Services (p. 18) is concerned with deriving specifications for composite services from the known specifications of the component services (pre- and post-conditions expressed as logic formulas), and with deciding which is the minimum part of these derived specifications that needs to be exposed to the user of the composition, so that, e.g., it can be used for constructing higher-level compositions.

Adaptation Based on Quality Prediction

Detecting ahead of time possible, probable, and/or imminent failure of service composition quality requirements (whether on the level of operational metrics, or related to higher-level business goals) during composition execution is the topic of two contributions presented in this deliverable.

Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitutions (p. 20) leverages machine learning techniques to identify, at a number of checkpoints in service composition execution, whether the composition is likely to violate its SLA requirements, which is usually expressed as a constraint on some QoS metrics that can be observed from monitoring. When such a likely violation is detected, the approach uses the techniques from Aspect-Oriented Programming (AOP) to substitute the current orchestration fragments with the new ones that have the chance of preventing the predicted SLA violation by changing the behavior of the composition.

Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Adaptation (p. 22) also relies on machine learning techniques, but with a different focus. Instead of focusing on low-level QoS metrics, here the authors look at the Key Performance Indicators (KPIs) of the business processes and try to infer influential factors on the process level that may lead to KPI violation on the business level. Several models (metrics, adaptation actions, constraints and preferences) are used to select the appropriate adaptation strategy based on adaptation requirements.

Context-Aware Service Compositions

Importance of context awareness for development of pervasive, dynamic and flexible service-based applications is the motivating factor behind several contributions presented in this deliverable. *Dynamic Composition of Pervasive Process Fragments* (p. 26) addresses the problem of dynamically creating service composition by putting together process fragments depending (among other things) on the context in which the composition needs to execute. That is done by using the globally available declarative specifications that define when a fragment can be used (relative to the context and other relevant requirements), and locally available imperative specifications that define its behavior.

Another interesting question is to predict context in which a service composition can find itself in some stage of execution. *Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction* (p. 28) addresses this problem by using a learning model at design time (which can be also updated at run-time), from which a service-based application retrieves prediction during execution.

A Context-Aware Framework for Business Processes Evolution (p. 24) covers several phases of the adaptation life-cycle: execution, analysis and evolution. In the execution phase, the framework manages the execution and adaptation of the system, and logs the relevant information for future use. In the analysis phase, the framework evaluates the quality of the observed KPIs, determines the need for evolution of a given process model, and identifies the contextual evolution problem that needs to be

solved. Finally, in the evolution phase, the framework computes the new process model that behaves better in terms of KPI and/or avoids violation of context-induced constraints.

Other Novel And Innovative Approaches

An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering (p. 30) aims at extending the existing management approaches to involve an environmental dimension. It proposes an architecture and a corresponding methodology that can help organizations define, measure, analyze and adapt service compositions with the green computing goal in mind.

Tweetflows - Flexible Workflows with Twitter (p. 32) introduces a human-mediated service-oriented layer on top of the publish-subscribe paradigm of the Twitter micro-blogging social network, to allow solicitation, provisioning, monitoring, and to facilitate adaptation of human- and computer-provided services in light-weight, mobile and dynamic project team environments.

Finally, *A Penalty-based Approach for QoS Dissatisfaction using Fuzzy Rules* (p. 33) uses fuzzy logic rules do deduce the amount of penalty that need to be compensated in case of an SLA violation, taking into the account the assessment of the degree (i.e., the amount) of violation and the characteristics of the service consumers

1.4 Key Research Challenges and Results

The S-Cube Integrated Research Framework (IRF) defines the following research challenges for the workpackage JRA 2.2:

Formal Models and Languages for QoS-Aware Service Compositions: This challenge deals with formal models and Languages for QoS-aware service compositions. The challenge is substantiated by the facts, that firstly, there are no formal models for service compositions available that take into account the QoS and behavioral characteristics of these compositions and secondly, that the formal models are extremely important to guarantee that the final result of a composition services possesses the required characteristics.

Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies: In the context of QoS-aware service compositions, the focus of this challenge lies on monitoring of quality characteristics of service orchestrations and service choreographies. As service compositions implement business processes and at the same time run on IT infrastructure, their quality characteristics are influenced by both process-level and infrastructure-level metrics. A holistic monitoring approach for quality characteristics of service compositions involves monitoring of service orchestrations in terms of both process-level and infrastructure level factors and in addition monitoring of quality characteristics across participants in service choreographies.

Analysis and Prediction of Quality Characteristics of Service Compositions: When monitoring of quality characteristics of service compositions reveals that KPIs do not meet their target values, users are interested in finding out the causes and the most influential factors in order to be able to adapt the composition to prevent those violations in a future. Analysis and prediction mechanisms for quality characteristics will be devised, which are integrated with the monitoring mechanisms and provide input to the adaptation framework on which quality characteristics to adapt.

QoS-Aware Adaptation of Service Compositions: Adaptations of Service Compositions driven by changes in the environment and in particular by changes in QoS characteristics still remains a major challenge in service-based applications. Mechanisms for enabling such adaptations will be developed, and the major drivers for adaptation will be defined. The influence of the BPM and SI layers of SBAs on the adaptation of SC must be taken into account to ensure consistency of the adaptation steps.

The relationship between the contributions and the above challenges are illustrated in the Table 1.1.

1.5 Relationship to Other Work Packages

As described above, the motivation for many approaches is closely associated with quality assurance, and the subject of the deliverable is very closely related to the work package WP-JRA-1.3: Quality Definition, Negotiation, and Assurance. The contribution titled *A Context-Aware Framework for Business Processes Evolution* (FBK) is also reported in the deliverable D-JRA-1.3.6 in that workpackage.

Contribution	Relationship of the Contributions to the Research Challenges in WP-JRA-2.2			
	Formal Models and Languages for QoS-Aware Service Compositions	Monitoring of Quality Characteristics of Service Compositions	Analysis and Prediction of Quality Characteristics of Service Compositions	QoS Aware Adaptation of Service Compositions
Awareness-Based Realizability Analysis Of Choreographies With Exception Handling Constructs (USTUTT)	✓			
Automatic Attribute Inference In Complex Orchestrations Based On Sharing Analysis (UPM)	✓		✓	
Towards Deriving Specifications for Composite Web Services (UoC, UPM)	✓			
Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitutions (TUW, USTUTT)			✓	✓
Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Adaptation (USTUTT, FBK)			✓	✓
A Context-Aware Framework for Business Processes Evolution (FBK)			✓	✓
Dynamic Composition of Pervasive Process Fragments (FBK)	✓			✓
Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction (UniHH)			✓	
An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering (USTUTT)				✓
Tweetflows - Flexible Workflows with Twitter (TUW)				✓
A Penalty-based Approach for QoS Dissatisfaction using Fuzzy Rules (UCBL, UPD, PolMi)		✓		

Table 1.1: Relationship of the contributions to the IRF challenges for WP-JRA-2.2.

Chapter 2

Mechanisms and Techniques for QoS-Aware, Coordinated Service Compositions

2.1 Awareness-Based Realizability Analysis Of Choreographies With Exception Handling Constructs

Publication reference: [27] attached on page 43.

Partners: USTUTT

Status: Technical Report.

2.1.1 Background

Service choreographies are a means of composing (web) services by specifying their roles, i.e. expected messaging behaviors, and the interactions among the roles. In the state of the art there are several paradigms for specifying service choreographies:

Interaction choreographies specify the message exchanges occurring among the participants in a choreography from a global perspective, e.g. using a workflow-like notation where activities represent message exchanges. Since the behaviour of the choreography is specified from a global perspective, the roles of the participants are specified *implicitly*. Interaction choreographies can be modeled with languages like Let's Dance [52], WS-CDL [16] and BPMN 2.0 Choreography diagrams [34].

Interconnection choreographies specify *explicitly* the roles played by the participants, and "wire" them by specifying how the messages generated by one role are consumed by another. Interconnection choreographies can be modeled, for example, using BPEL4Chor [8] or BPMN 2.0 Collaboration diagrams [34].

Artifact-centric choreographies (see e.g. [26]) focus on how the lifecycles of business objects are manipulated by the participants through message exchanges.

The global perspective adopted by interaction choreography modeling gives raise to the issue of *realizability*. An interaction choreography is realizable it is possible "...to automatically extract from the choreography the behavioral skeletons of the participants so that the concrete implementations, built on the basis of these skeletons, are guaranteed to satisfy the choreography specification." [17]

2.1.2 Problem Statement

Realizability is a fundamental property of a choreography model [28], as it guarantees that the choreography can be actually realized faithfully by its participants. While the state of the art comprises a large body of theoretical works on realizability (for example [17, 44]) and methods for verifying it (see e.g. [5, 25, 39]), there is still only limiteds understanding of how to *alleviate* the *realizability defects* of an interaction choreography by modifying it so to make it realizable.

The first step towards a method for alleviating realizability defects of a choreography requires a method that provides meaningful diagnostic information of its realizability defects. In particular, since the unrealizability of a choreography is due to the lack of synchronization among its participants (see e.g. [44, 39]), it is vital for correcting realizability defect to have diagnostic information that pin-points: (1) in which point of the choreography is more synchronization required, and (2) why the participants that should be synchronized, are in fact not.

2.1.3 Contribution Relevance

The realizability analysis proposed in this work provides precise diagnostic information of realizability defects on interaction choreographies modeled with ChorTex, a process algebra-based choreography language based on Chor [51] (which, in turn, is related to WS-CDL).

2.1.4 Contribution Summary

In this work we propose a novel method for the realizability analysis of interaction choreographies modeled with ChorTex. The method is based on the concept of awareness, i.e. using static analysis for plotting the participants' "perception" on the global state of a choreography enactment. By transforming a ChorTex choreography to a Control Flow Graph (CFG), and annotating the CFG's nodes with the awareness of its participants using a fixed-point algorithm, the realizability analysis provides very precise information on which participants have perception of which states of the global enactment. We verify the realizability of the choreography by verifying *awareness constraints* that, if satisfied, guarantee the choreography's realizability. If the choreography is unrealizable, the unsatisfied awareness constraints given precise and localized information on the synchronization shortcomings of one or more of its participants. Moreover, by traversing the CFG backwards, it is easy to discover at which point in the enactment, the participant has lost the necessary synchronization.

2.1.5 Contribution Evaluation

Formal proofs of correctness and convergence of the proposed method.

2.1.6 Relation to the Research Framework

The realizability analysis presented in this work is meant to support the design and evolution of service choreography by (1) allowing designers to verify the realizability of their choreography specifications (JRA-2.2), and (2) lay the foundation to a method for proposing changes that would solve realizability defects in choreographies, i.e. to steer the adaptation of service choreographies in order to make them realizable (JRA-1.2).

2.1.7 Conclusions

Using the diagnostic information that our realizability method provides, we intend to investigate how to identify change alternatives that would solve the realizability defects. Finally, we are investigating how to use the concept of *monitoring contracts* (see e.g. [49]) as an alternative method of alleviating realizability defects in choreographies when adapting the choreography specification is not an option.

2.2 Automatic Attribute Inference In Complex Orchestrations Based On Sharing Analysis

Publication reference: [13] attached on page 116.

Partners: UPM

Status: Presented at the 2011 IEEE International Conference on Services Computing (SCC 2011), Washington DC, July 4-9, 2011.

2.2.1 Background

Service compositions allow expression of a higher level or cross-organizational computation processes based on the existing component services as the building blocks. They are often given in the form of orchestration and defined using a workflow that specifies activities and the control and data dependencies between them. Complex activities include branches, loops and parallel flows, while simple activities denote single steps in the workflow. Some of the latter are executed locally, while others invoke component services, which, in general, may be provided and managed by different organizations. Reasoning about properties of data that is fed as an input to an orchestration, read / written and circulated between its activities, and returned as the result, can be useful in several phases of service lifecycle, such as composition design, adaptation through fragmentation and distributed enactment, and refactoring.

2.2.2 Problem Statement

The contribution addresses the problem of automated inference of attributes (user-defined properties) of data and activities inside a service orchestration that may include complex control and data structures and dependencies. The attributes are chosen by the user (designer) to reflect the relevant aspects from the application domain, such as privacy levels, data source or information content. Starting from the known user-defined matrix of attributes of the inputs to the orchestration and its structure, the goal is to deduce the attributes of the intermediate data items, data read by individual activities, as well as of the end results. That is a generally undecidable problem in presence of complex control structures, such as loops, and therefore, without restricting the class of admissible orchestrations to acyclic cases, the inference can be allowed to produce approximate, but safe (conservative) results, where potential attributes (i.e., those that cannot be decisively ruled out) are included.

2.2.3 Contribution Relevance

In the previous work [14] (presented as a contribution to the previous deliverable [12]), the authors presented the approach in which an input lattice of abstract data properties, such as data access / privacy levels, can be fed to a Horn clause representation [24] of a workflow to produce, based on the notion of *data sharing*, an output sharing lattice that can be used to identify workflow fragments by grouping together activities that share the same position in that lattice. In the current contribution, the authors present a sketch of a complete automated round-trip solution where a designer uses simple (Boolean) characterization of the inputs, which is internally transformed into an underlying input lattice structure, and is, at the end of the analysis, presented with a Boolean matrix that assigns the user-chosen input attributes to the workflow activities, intermediate data items and the resulting message. The approach works with orchestrations that have complex, real-life control structures which may include loops, branches and parallel flows.

2.2.4 Contribution Summary

The proposed approach uses the notions from Formal Concept Analysis (FCA) to transform a tabular object-attribute specification of the workflow inputs into a *context lattice* [11]. At the same time, the workflow definition is mechanically transformed to the form of a *logic program* (a series of Horn clauses) [24], which are then subjected to *sharing analysis* [31]. The Horn clause translation models both the data dependencies between activities, and control dependencies, such as branches, loops and parallel flows. Data items and activities are represented with logic variables. The sharing analysis operates in an abstract (conservatively approximate, but decidable) *sharing and freeness* which uses the input concept lattice to set up the initial (abstract) sharing relationship between the logic variables that represent the workflow inputs. The result of sharing analysis, the resulting sharing lattice, is then used as a concept lattice that is processed to recover characterization of the intermediate data items and activities in terms of the user-defined attributes. The final result is presented to the user as a Boolean object-attribute table.

2.2.5 Contribution Evaluation

The applicability of the proposed approach was illustrated on the case of distributed enactment of a medical workflow. The workflow used the inputs that identify a patient, as well as the databases of medical records and previously prescribed medications. In this setting, the user-defined attributes described the information content of the inputs, while the workflow (which involved parallel splits and joins, branches and loops, as well as sub-workflow components) was aimed at distributed enactment between the central health organization and its external partners (such as examiners, pharmacists and clerical service providers). The proposed approach was used to infer the information content of the intermediate data items in the workflow, as well as the content of information "seen" by each individual activity. The process was divided into fragments that were either executed centrally, or sent to a partner, based on data disclosure policies and the inferred information content. Other applications, such as robust top-down development and specification refinement were also considered.

2.2.6 Relation to the Research Framework

This contribution is directly related to the WP-JRA-2.2 research challenge *Analysis and Prediction of Quality Characteristics of Service Compositions*, where the quality aspect considered falls into the class of "data quality," such as privacy and security. Its application to fragmentation (as a form of adaptation) is also related to the research challenge *QoS Aware Adaptation of Service Compositions*.

2.2.7 Conclusions

The contribution shows how an FCA-based characterization of input data to a workflow can be enriched to include intermediate data items and internal activities. These are annotated with attributes which are inferred from emergent properties of the workflow which stem from the workflow structure and relationships between input data. That task can be automated by translating (a) an initial FCA into a lattice from which sharing conditions are derived and (b) the workflow structure into a logic program. Then, (a) and (b) are subjected to a sharing analysis, and the results are mapped back to a resulting lattice and that to a resulting context, whose information can be used as a starting point for a number of other tasks. We have illustrated this methodology with a worked example.

As future work, the authors plan to address the development of automatic translations from common business process specification languages (BPEL, XPDL, YAWL, etc.) into logic programs amenable to sharing analysis in order to further test and refine the techniques proposed herein. Besides, they plan to explore other applications of the concept of sharing to services, aiming not only at (local) data sharing between activities, but also looking towards the representation of stateful service conversations and quality aspects of services.

2.3 Towards Deriving Specifications for Composite Web Services

Publication reference: [1] attached on page 124.

Partners: UoC, UPM

Status: Conference, Submitted

2.3.1 Background

Service composition enables service-based systems to be built using accepted engineering principles such as service reusability and composability with the aim to provide value-added services that achieve functionality otherwise unattainable by atomic services. In order to fully achieve these goals, composite services should be available to consumers in the same way as atomic services are, abstracting away complex details of the way participating services are orchestrated to achieve the required functionality. This allows service consumers to invoke services regardless of the way they are implemented. This can be accomplished by providing formal specifications of composite services which reveal to the end user the minimum information required to understand the functionality offered, often by describing the inputs, outputs, preconditions and effects (collectively known as IOPEs) of the composite service. The composite service specifications should be based on existing specifications and descriptions of the services taking part in the compositions.

2.3.2 Problem Statement

While existing service description frameworks attempt to describe service compositions using a variety of composition models ranging from orchestrations to choreographies to Finite State Machines, no attempt (to the best of our knowledge) has been made to handle the problem of automatically producing specifications for a composite service, based on the specifications of the participating services. The same is true for automated Web service composition approaches: while each of them offers a way of automatically or semi-automatically producing the composition schema, the control flow and data flow of the composite service, none attempt to derive a complete specification IOPEs that should be provided to the service consumer.

Apart from the convenience they may provide to service consumers, specifications can be an invaluable tool for service providers. Similarly to the case of programming specifications, service specifications could be used as a basis to construct a service based on a set of requirements agreed upon by the parties involved, or to check that some existing specification meets a set of requirements. Specifications also play a major role in verification techniques as well as in the evaluation of the results of service adaptation or service evolution.

Composite specifications also offer great assistance when one attempts to deduce whether a set of services can actually be composed in a meaningful way. During the process of creating the composite specifications, inconsistencies may be detected between preconditions and/or postconditions of the participating services, rendering that particular set of services not composable. Such problems can be prevented before the composite service is delivered to the end user, by replacing the service or services that cause the inconsistencies.

2.3.3 Contribution Relevance

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

bottom-up approach. The proposed process keeps complexity at a minimum (as opposed to trivially including the complete specifications of participating services in the composite service specification) while at the same time retaining the complete set of knowledge that should be provided to the end user.

2.3.4 Contribution Summary

The composite specification should explicitly state all conditions that must be true before the execution of the whole composite service, as well as all conditions that are true after a successful execution. While we have preconditions and postconditions for each participating service, there is no obvious way of deciding which part of them will be included in the composite specification. Exposing the conjunction of the preconditions and postconditions of the participating services ignores the way the services are orchestrated and therefore the composed specification may be cumbersome, if not incorrect.

We propose a derivation process that is based on structural induction and attempts to construct the composite specification using a bottom-up approach. In order to achieve this, we formulate the derivation for all fundamental control constructs, namely sequential composition, AND-Split/AND-Join, OR-Split/OR-Join and XOR-Split/XOR-Join. By deriving preconditions and postconditions for these constructs, we can derive specifications for any composite service that includes such constructs, by considering the composition schema. The derivation process begins by examining the construct deeper in the schema and gradually moves its way upwards, till the whole composition schema is considered.

We also handle loop specification by deriving preconditions and postconditions based on the loop invariant. Generating loop invariants is an active and open research area and we provide precise details on what is necessary from approaches in this area for our derivation process to be valid. Finally, the case of handling asynchronous execution is addressed by employing the static single assignment form (SSA) in order to make sure that preconditions are evaluated in the context of the request and not the response.

2.3.5 Contribution Evaluation

The derivation process is applied to the E-Government case study of S-Cube [32] involving the request of paid-for, government-issued documents with or without certification by citizens. We explore a service evolution scenario, where a composite service implementing part of the document issue process is evolved to a new orchestration covering the complete process, satisfying a revised set of requirements. In order to check the conformance of the evolved service with the new specification, a specification for the new orchestration is derived using the method described in the paper. The resulting specification offers the complete set of preconditions and postconditions that formally describes what the composite service is supposed to provide and under what circumstances.

2.3.6 Relation to the Research Framework

This contribution is closely related to the research challenge for formal models and languages for QoS-aware service compositions, as it provides an automatic process of deriving formal specifications based on first-order logic that describe composite services. Moreover, there is a relation to challenges related to service adaptation, such as QoS Aware adaptation of service compositions, since the derivation process can be used to re-specify a composite service, after its composition schema and possibly some of its participating services have been adapted.

2.3.7 Conclusions

We proposed an approach for inferring composite service specifications given the specifications of the services participating in the composition (in the form of sets of preconditions and postconditions) and the composition schema. The approach attempts to construct the specification by using structural induction based on derivation rules defined for all fundamental control constructs, including loops as well

as supporting asynchronous execution. The resulting specification can be used to formally describe the composite service in terms of its preconditions and postconditions without requiring any knowledge of the internals of the composition, allowing for an actual "black box" view of the whole process. Such a formal description can be then employed to check whether a composition satisfies the requirements set by the requester, especially in cases of service evolution and service adaptation.

The nature of the proposed approach facilitates a possible implementation: structural induction lends itself to be written as a recursive algorithm. Hence, it would be straightforward to create an automated process that takes a set of service specifications and a composition schema and produces the specification for the composite service of the schema.

Future work includes exploring specification simplification, which becomes important as composite services become more complex. We plan to look into the work of Douglas Smith [43] on simplifying precondition formulas. An implementation is also in the works, in order to evaluate the approach in terms of effectiveness with respect to time. Finally, the work is planned to be integrated in a general service specification framework that will handle issues raised by the frame problem and related ones.

2.4 Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitutions

Publication reference: [21] attached on page 134.

Partners: TUW, USTUTT

Status: Conference, Accepted

2.4.1 Background

In a service-oriented architecture, service providers realize business processes through a service composition which orchestrates services running on a service infrastructure. They offer a service to service consumers agreeing on its QoS characteristics in a service level agreement (SLA).

2.4.2 Problem Statement

For a service provider it is desirable to be able to *predict* whether an SLA will be violated. In addition, after an SLA violation is predicted, the goal is to proactively adapt the composition in order to prevent the predicted violation. Thereby, it is often not sufficient to substitute services in the composition, but it is often needed to change the process logic in terms of composition fragments.

2.4.3 Contribution Relevance

The presented approach is based on and extends several previous S-Cube papers. [50] has shown how machine learning techniques can be used for KPI and SLA dependency analysis, i.e. finding out the influential factors of SLA violations. Here, we use a similar learning approach focusing on prediction. [22] has introduced the concept of check points which we reuse in the approach. [20] has dealt with prevention of SLA violations by integrating monitoring, prediction and adaptation in a similar fashion, however focusing just on service substitution as the adaptation mechanism. We extend that approach by enabling the substitution of arbitrary process logic in terms of composition fragments.

2.4.4 Contribution Summary

At design time, first a set of *checkpoints* is defined in the service composition where the prediction and potential adaptation should take place. Then, a set of fragment alternatives is modeled by the user and assigned to the checkpoints. A *fragment* is an (alternative) implementation of a part of business logic in the target composition. It is a standalone and typically "incomplete" composition part focusing on a certain part of business logic and is *linked* into the original composition via join points. The linking is performed at design time by the user. Therefore, one specifies at which points in the original composition the control flow changes from the target composition to the fragment and vice versa. The join points in the fragment are defined via *virtual activities* denoting the START, END, or TRANSPARENT parts of the fragments. In addition to the functional part, a fragment definition also contains an *impact model* which defines how this fragment affects composition performance metrics. This allows later to select appropriate fragments which will likely prevent the predicted violation. The impacts can be derived from SLAs or from history measurements. Finally, one can also specify dependencies between fragments, e.g. one fragment requiring another one.

At process runtime, for each checkpoint a prediction model is learned based on history data using machine learning techniques. Whenever a composition instance passes the checkpoint, the monitored data of that instance is used as input for the prediction model. If an SLA violation is predicted, the adaptation is triggered. A set of adaptation alternatives is identified based on the impact models by repeating the prediction for each potential adaptation alternative (all possible fragment combinations are enumerated) as if adaptation has taken place. From the alternative adaptation alternatives, the alternative whose predicted SLA value is closest to the target value is weaved into the target composition.

2.4.5 Contribution Evaluation

The approach has been implemented based on Windows Workflow Foundation technology and has been experimentally evaluated. Experiments show that dynamic weaving introduces only a very small overhead (in [45:80] ms) which could be relevant only for very short running processes.

2.4.6 Relation to the Research Framework

This contribution aims to provide techniques and a development process to design *QoS-aware adaptable service compositions* which is one of the main tasks of JRA-2.2. It covers the research challenges "Analysis and Prediction of Quality Characteristics of Service Compositions", "Comprehensive and integrated adaptation and monitoring principles, techniques, and methodologies", "QoS Aware Adaptation of Service Compositions", and "Proactive Adaptation and Predictive Monitoring". It therefore connects the domain layer *Service Composition & Coordination* with the cross-cutting concerns *Engineering and Design*, *Quality Definition, Negotiation and Assurance*, and *Adaptation and Monitoring*. Regarding the S-Cube reference life-cycle, the approach covers the life-cycle states *Requirements Engineering and Design* and *Construction* by explaining how fragments and their linking are to be modeled at design-time, and then focuses on the runtime states from *Operation and Management* to *Adaptation Enactment*.

2.4.7 Conclusions

The presented approach deals with runtime adaptation of service compositions for preventing SLA violations. It is based on previous S-Cube work on monitoring and runtime prediction of SLA violations. The main contribution is a new adaptation technique which substitutes composition fragments dynamically at runtime. At design time, composition fragments are modeled separately and are explicitly linked into the original composition using virtual activities. At runtime, at predefined checkpoints they are then weaved into the original composition if an SLA violation is predicted. Future work includes taking into account several service level objectives at the same time and the cost of adaptation.

2.5 Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Adaptation

Publication reference: [48] attached on page 149.

Partners: USTUTT, FBK

Status: Journal, Submitted

2.5.1 Background

Service-based applications (SBAs) realize business processes through a service composition which orchestrates services running on a service infrastructure. An important aspect when managing such applications, is to ensure that they meet certain business goals, typically expressed via a set of Key Performance Indicators (KPIs). KPIs specify target values on the time, cost, and quality dimensions of business processes.

2.5.2 Problem Statement

KPIs are typically continuously monitored at process runtime using business activity monitoring techniques. If monitoring shows bad results, i.e. KPI targets are often violated, then one needs to analyze the influential factors which lead to those violations. In complex business processes, the dependencies of KPIs on lower-level metrics are neither explicit nor easy to reveal. In addition to finding out the reasons for KPI violations, it is desirable to be able to *predict* whether a KPI target will be violated. This would allow to proactively adapt the process in order to prevent a predicted violation. Thereby, one should take into account that typically several competing KPIs can be specified (e.g., quality vs. cost), which has to be addressed during prediction and adaptation. The overall goal is to create an approach which integrates the monitoring, analysis, prediction and adaptation phases to achieve self-adaptable KPI-aware service compositions.

2.5.3 Contribution Relevance

The presented approach is based on and related to several previous S-Cube papers. [50] has shown how decision tree learning can be used for KPI dependency analysis, i.e. finding out the influential factors of KPI target violations. Here, we use the same learning approach with the focus on using the KPI dependency trees for prediction and adaptation purposes. In [18], we have already discussed on a higher-level how adaptation requirements can be extracted from dependency trees and how to derive adaptation strategies. That work has been substantially extended, implemented and evaluated in the current approach. [22] has introduced the concept of check points which we reuse in our approach. [20] and [21] (presented in Section 2.4) deal with prevention of SLA violations by integrating monitoring, prediction and adaptation in a similar fashion. The prediction thereby focuses on numerical metrics using machine learning techniques such as artificial neural networks. In our approach, we predict nominal-valued metrics using decision trees which is a white-box model explaining explicitly dependencies between the KPIs and lower-level metrics and thus allowing extraction of adaptation requirements. We also support prediction and adaptation in respect to several competing KPIs.

2.5.4 Contribution Summary

At design time, several models are created which are then used as input to the runtime monitoring and adaptation phases. The *metrics model* contains the KPI definitions and the lower-level metrics which are needed for dependency analysis and prediction. The *adaptation actions model* contains the available

adaptation actions (AAs) which can be used to adapt the process, e.g. service substitution, change of a process variable value, skipping of an activity etc. In addition, an AA specifies its impact on a set of metrics which is later needed for deciding whether this AA actually helps to prevent the KPI violation. The *check point model* defines the check points in the process where the prediction and potential adaptation should take place. Finally, the *constraints and preferences model* defines (i) constraints on metrics and KPIs which have to be satisfied when performing adaptation, (ii) preferences in terms of weights on the KPIs and metrics which allow selecting an adaptation strategy in case of several alternatives.

At runtime, the metrics model is used in the monitoring phase to calculate KPIs and metric values based on runtime events. After a certain number of process instances has been executed, for each check point and each KPI of that check point, a KPI dependency tree is learned. In the prediction phase, the process instance is halted at a check point and the KPI dependency trees are used to predict the KPI classes for that process instance by inserting the available metric values at the check point into the trees. The prediction result is a decision tree (derived from the learned KPI dependency tree) which shows the predicted KPI classes in relation to the metrics affected by available adaptation actions. Form the derived instance trees we extract adaptation requirements (conjunctions of metric value predicates) which specify how we have to improve certain metrics. In the subsequent step, we then select and combine available adaptation actions which satisfy the adaptation requirements, thus creating adaptation strategies. One adaptation strategy is finally selected based on the specified preferences model using simple additive weighting as part of multiple attribute decision making.

2.5.5 Contribution Evaluation

The approach has been implemented and experimentally evaluated based on a purchase order processing scenario. The experiments show that the prediction and adaptation time together are below a second thus making it only a performance factor for short running processes. More importantly, we have shown that the number of KPI violations has been considerably decreased with the use of our framework. The prevention effectiveness mainly depends on (i) check point positioning, i.e., the later the check point, the higher the prediction accuracy, however the lower the number of available adaptation actions, (ii) conformance of actual metric effects of chosen adaptation actions to the modeled (estimated, guaranteed) metric effects as specified in the adaptation actions model; (iii) preferences model, i.e. provided weights on the different KPIs and metrics.

2.5.6 Relation to the Research Framework

This contribution aims to provide techniques and a development process to design *QoS-aware adaptable service compositions* which is one of the main tasks of JRA-2.2. It covers the research challenges "Analysis and Prediction of Quality Characteristics of Service Compositions", "Comprehensive and integrated adaptation and monitoring principles, techniques, and methodologies", "QoS Aware Adaptation of Service Compositions", and "Proactive Adaptation and Predictive Monitoring". It therefore connects the domain layer *Service Composition & Coordination* with the cross-cutting concerns *Engineering and Design*, *Quality Definition, Negotiation and Assurance*, and *Adaptation and Monitoring*. Regarding the S-Cube reference life-cycle, the approach covers the life-cycle states *Requirements Engineering and Design* and *Construction* by explaining the different models which have to be created at design-time, and then focuses on the runtime states from *Operation and Management* to *Adaptation Enactment*.

2.5.7 Conclusions

The presented approach integrates monitoring, KPI dependency analysis, prediction and proactive adaptation to realize QoS-aware adaptable service compositions. The monitoring and adaptation are thereby currently focused on the service composition layer. Future work consists of extending the approach

towards cross-layer monitoring and adaptation, e.g. taking into account the impact of the service infrastructure QoS metrics on the KPIs and performing adaptations on service infrastructure level accordingly.

2.6 A Context-Aware Framework for Business Processes Evolution

Publication references: [3] (attached on page 165) and [4] (attached on page 174).

Partners: FBK

Status: Workshop Paper presented at EVL-BP 2011 and Demo paper accepted at ICSOC 2011

2.6.1 Background

Run-time adaptability 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. Process evolution exploits the information on process variants to identify the best performing recurring adaptations and adopt them as general solutions in the process model. However, process variants are strictly related to specific execution contexts and cannot be adopted as general solutions. 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 adaptation need. We demonstrate the benefits of our approach using a car logistics scenario.

2.6.2 Problem Statement

Adaptation needs of dynamic business processes may be triggered by specific cases to be handled, unexpected situations depending on environmental conditions, or changing requirements.

This 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. In such a dynamic environment, the process models cannot remain unchanged; the short-term adaptations applied to process instances should be used to derive long-term changes in the process models. Providing support for process model evolution is becoming one of the main requirements for managing the lifecycle of dynamic processes. 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., [38, 23]) 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", it is good for 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. Moreover, 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.

2.6.3 Contribution Relevance

To overcome the limitations, presented in the previous section, we present a context-aware evolution framework that, instead of searching for recurring process changes, searches for recurring adaptation needs. We describe the modeling artifacts of our evolution framework, as well as their lifecycle. Finally, to illustrate the role of these artifacts, we use a car logistics scenario.

2.6.4 Contribution Summary

The framework that we propose is able to cover the following phases of an evolution life-cycle, namely (1) *execution phase*, (2) *analysis phase*, and (3) *evolution phase*. During the first phase, the evolution framework proposed, 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.

2.6.5 Contribution Evaluation

The evolution framework has been implemented as a part of the CAptEvo prototype [4], a tool of the ASTRO project [1], that integrates sophisticated techniques for managing the execution, adaptation, and evolution of context- aware business processes. Finally, to demonstrate the CAptEvo framework in action, we use a real-world scenario from the domain of logistics.

2.6.6 Relation to the Research Framework

This contribution spans over several challenges of different workpackages. In particular, it is closely related to the JRA-2.2 research challenge of *Analysis and Prediction of Quality Characteristics of Service composition* and *QoS Aware Adaptation of Service Compositions*, since it provides techniques to automatically improve the process composition on the basis of QoS performances. The approach also contributes to the *Comprehensive and integrated adaptation and monitoring principles, techniques, and methodologies* challenge of JRA-1.2, and to *Run-time Quality Assurance Techniques* and *Quality Prediction Techniques to Support Proactive Adaptation* challenges of JRA-1.3. Regarding the S-Cube lifecycle, the proposed framework covers all the phases related to monitoring, adaptation and enactment, namely *Operation & Management*, *Identify Adaptation Need*, *Identify Adaptation Strategy*, and *Enact Adaptation*.

2.6.7 Conclusions

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

[1] www.astroproject.org

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.

2.7 Dynamic Composition of Pervasive Process Fragments

Publication reference: [42] attached on page 176.

Partners: FBK

Status: Conference paper (accepted) and journal paper (on-going)

2.7.1 Background

Applying the SOA paradigm to pervasive systems (Internet of Things) is a hot reaserach topic that introduces several interesting challenges for the Service Community. Pervasive systems change their behavior, reconfigure their structure and evolve over time reacting to changes in the operating conditions, so to always meet users' expectations. This is fundamental since those systems live in distributed and mobile devices, such as mobile phones, PDAs, laptops etc., thus their environment may change frequently. Also, user goals and needs may change dynamically, and systems should adapt their functionalities accordingly, without intervention from technicians. In order to achieve the required degree of flexibility and dynamicity a SOA based solution must cope with the fact that services must be context-aware, self-configurable and adaptive.

2.7.2 Problem Statement

The ability to model partial and incomplete knowledge about the processes and to dynamically compose them into complete executable processes is a key feature of pervasive applications, where the actual process to be executed depend on the specific context and on the available services. Process fragments [10] represent a tool for modeling incomplete and local process knowledge. The knowledge is incomplete since the modeler is allowed to leave gaps in the process specification. Further, it is local, since the availability and usability of the fragments is determined by context (e.g., location, time, situation, people). For example, the process fragment execution may be bound to a certain location. We propose a solution for automatically compose process fragments according to predefined complex flow goals and taking into account fragments availability and context properties.

2.7.3 Contribution Relevance

Several approaches have been proposed to dynamically compose services [47, 37, 19], however, they cannot be applied to pervasive domains since they do not deal with the context aspect. Moreover, all these approaches represent the process goal either through an imperative or a declarative language. Imperative process models focus on the way to achieve the goal, and assume that the environment is stable. Declarative process models specify the goal through constraints which approximate the desired behavior. Such descriptions are suitable for frequently changing environments, but cannot be executed completely automatically. We use both process description techniques to achieve a greater flexibility with respect to the environment. The declarative part corresponds to goals, and is globally available. The imperative, concrete part corresponds to pervasive process fragments, and is available only locally.

2.7.4 Contribution Summary

We propose an approach for composing pervasive process fragments according to complex goals. Our approach is based on previous results from Web service composition [37], and in particular on the work of [2] , which addressed the problem of Web service composition with complex composition requirements. The problem of fragment composition is conceptually different from Web service composition, since the components are not orchestrated, but integrated into a complete flow model. Due to this difference, important issues in Web service composition, such as the need for Web services to communicate via message exchange, do not appear in fragment composition. On the other hand, pervasive fragment composition introduces new problems. For example, the fragments can overlap, or can have gaps in the specification, and the composition task must take into account context properties. To deal with these differences, we have significantly modified and extended the approach in [2]. Our approach is based on a three-layer representation, where the first layer captures the specific context knowledge required for the composition task, the second layer describes the abstract processes in terms of the goals it should achieve, and the third layer is the concrete, context-dependent part of the process definition, represented using fragments. We present a complete model that allows to relate context properties, goals, and fragment models, and a mechanism for composing fragment by encoding object diagrams, goals and fragments into an AI planning problem.

2.7.5 Contribution Evaluation

We implemented the approach into a prototype tool. The tool translates object diagrams (XML), process fragments (BPEL extension) and goals into a planning problem. The planning problem is given as input to WSYNTH, one of the tools in the ASTRO toolset (http://www.astroproject.org). The output returned by WSYNTH is the planning controlled composition, which we then translate back to executable BPEL. We evaluated our tool on a logistic scenario (handling and delivering boxes in an airport) entailing a high level of complexity and pervasiveness. We consider two features specific to fragment composition. First, there is a tradeoff between designing fragments with a large number of activities (a higher burden on the designer) and with a small number (longer composition time). Second, the set of available fragments can contain more fragments then actually necessary for composition. We observe that larger fragments lead to a significant speedup. Using fragments of two or more activities can lead to a time increase of 20% in the worst case, and a decrease of up to 95% in the best case. For this domain involving complex composition problems, larger fragments provide a clear benefit in terms of composition time. In the second experiment, we test how the number of available fragments influences the performance. For this purpose, we vary the total number of fragments, while keeping constant the number of fragments actually composed. As expected, the composition time grows exponentially in the number of fragments, due to the hardness of the planning problem. However, this can be solved using existing selection techniques to reduce the number of component fragments.

2.7.6 Relation to the Research Framework

This contribution is closely related to the JRA-2.2 research challenge of *Formal Models and Languages for QoS-Aware Service Compositions*, since it provides a framework for modeling partial, local, and context-aware process knowledge through fragments. It is also related to the research challenge of *QoS Aware Adaptation of Service Compositions*, since it provides an automatic way of composing at run-time the set of available, and context/goal -specific fragments to obtain a context-aware executable process. Regarding the S-Cube life-cycle, the proposed modeling framework (domain objects, pervasive fragments, complex goals) can be mapped to the life-cycle phase *Requirements Engineering and Design Construction*. While the dynamic fragment composition contributes to the *Enact Adaptation* phase, since compositions can be automatically derived at run-time considering the execution context.

2.7.7 Conclusions

The approach presented in this Section allows to compose pervasive process fragments according to context and goals. We first specify the properties of the context using object diagrams, and use these properties to define process goals and to annotate the fragments. We then encoded object diagrams, goals and fragments into an AI planning problem. The result of composition is a complete adaptable pervasive process which satisfies the goals and corresponds to a synthesis of the pervasive fragments. An on-going work is the complete formalization of the proposed framework and its evaluation on a logistic scenario in the automotive domain.

2.8 Designing Future-Context-Aware Dynamic Applications with Structured Context Prediction

Publication reference: [53] attached on page 184.

Partners: UniHH

Status: Accepted for publication in Software—Practice & Experience (SPE)

2.8.1 Background

Middleware support for advanced service-based applications includes dealing with heterogeneous systems and dynamic changes of execution environments. If applications are *context-aware* they are able to detect such changes at runtime and react to them accordingly. Following Dey and Abowd, *context* is defined as any information characterizing the situation of surrounding and situational entities relevant to an application and its users [9]. In consequence, the ability to obtain, to process, to manage and to provide relevant context information describing the user's environment and situation has become one important requirement for such systems. In addition to that, the *prediction of future context* is another important step for enabling devices and applications to also proactively support the user or to enable the desired automatic execution of his tasks even in dynamic environments [30]. However, the design of reusable middleware support for such *future-context-aware* service-based applications is still challenging – since a supporting prediction system has to be both generic and, at the same time, as efficient and accurate as possible.

2.8.2 Problem Statement

In summary, a general applicability of a context prediction system imposes several principal requirements: First, the system should support a prefereably wide range of applications and diversity of exploitable contexts [6] in order to maximize reusability. Furthermore, there are individual differences between users [15] which can also change continuously [41]. Thus, the system has to adapt to the individual user at runtime by learning about the characteristics and regularities which determine the future context [15, 41].

Besides being generic and adaptive, the prediction system should be able to produce customized predictions for different scenarios in a reliable and satisfying way, i.e. as accurately and efficiently as possible. Especially mobile devices often suffer from a lack of resources [40], so that the corresponding requirement for efficiency makes this trade-off even more challenging. Finally, the motivation to support application developers implies a preferably low effort for them.

2.8.3 Contribution Relevance

Many existing approaches aim to support predictions for a particular domain or a specific context criteria. Only few approaches explicitly address the design of a *prediction framework* to support the development of future-context-aware applications in general. Here, the approaches of Mayrhofer [29], Sigg [41] and Petzold [36] are considered to be major contributions towards generic context prediction. However, the analysis of existing work has shown that there is still no approach which is sufficiently generic and offers high accuracy and efficiency at the same time. Based on the experiences with existing approaches this contribution introduces the approach of *Structured Context Prediction (SCP)* which addresses open issues and therefore presents an alternative to existing context prediction frameworks.

2.8.4 Contribution Summary

The approach of *Structured Context Prediction (SCP)* realizes a generic prediction system which can be utilized in order to develop future context-aware applications or enhance existing applications with the ability to make assumptions about future situations. It is based on two fundamental principles derived from the preceding analysis of existing approaches: First, knowledge about the application domain is used as valuable information incorporated by the application developer at design time or at runtime. The second principle is a hybrid application of multiple, exchangeable prediction methods. Thus, methods which are appropriate to ensure accuracy and efficiency of domain-specific predictions can be selected and combined by the application developers respectively. Based on that, so-called *prediction models* can be created and incorporated at design time of the application or can be distributed at runtime. Additionally, an adaptation to the individual user is achieved by adaptive online-learning as the default learning mechanism.

The knowledge about an application domain is described as a *prediction model* which specifies the way predictions have to be performed. Basically, it assigns a method to each relevant variable in order to predict its value. As input, the method uses the values of other variables – which are again predicted by their own methods or which are known (e.g. measured by a sensor or distributed at runtime). The main part of a prediction model specifies how the methods and respective variables are connected. This part is called *prediction net*. A prediction net specifies that the value of a variable at a specific point of time is predicted using the values of other variables at the same point of time or earlier.

From the perspective of an application developer, the general procedure of using the prediction system consists of two parts: The first part is determined by the development of the prediction model at design time. The second part is the retrieval of predictions by the respective application at runtime. Runtime learning is executed automatically so that most aspects of learning are hidden from the application and from the developers.

2.8.5 Contribution Evaluation

The proposed approach has been evaluated in a mobile environment, where the successful execution of service-based business processes depends highly on the availability of required services. In order to access such unsteadily available services and exploit local functionalities, processes can be migrated to other mobile or stationary devices. In such a scenario, it is advantageous to guide the migration of a mobile business process by predicting future service availability.

This scenario has been used in order to evaluate the proposed approach in a case study by creating an corresponding prediction model and by examining realistic context data about the behaviour of a user and its mobile device spanning an interval of seven days.

2.8.6 Relation to the Research Framework

This contribution aims to provide techniques and a development process to design or enhance future-context-aware service-based applications. It therefore connects the domain layer *Service Composition*

& Coordination with the cross-cutting concerns *Engineering and Design* and – in parts – *Quality Definition, Negotiation and Assurance*. Regarding the S-Cube reference life-cycle, the proposed procedure of developing a future-context-aware application can be mapped to the life-cycle states *Requirements Engineering and Design* and *Construction*. The contribution also introduces techniques for the actual context prediction, which are utilized in the phase *Operation & Management*

2.8.7 Conclusions

As a further step towards realizing dynamic pervasive environments, the approach of Structured Context Prediction proposes and realizes a context prediction system which is generic and provides potential for high accuracy and efficiency at the same time. In consequence, this approach provides a new combination of characteristics compared to existing approaches. In contrast to previous approaches, the composability of prediction methods and the integration of domain-specific knowledge as proposed here enable support for a wide range of new (service-based) applications and thus allow integration into existing middleware frameworks.

2.9 An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering

Publication reference: [33] attached on page 204.

Partners: USTUTT

Status: International Conference on ICT as Key Technology for the Fight against Global Warming - ICT-GLOW 2011, accepted and published.

2.9.1 Abstract

Sustainability and responsible resource exposure has become a major issue in everyday life. Government, customers, and increasing social responsibility force more and more organizations to positively optimize their environmental impact towards a better, livable planet. In this paper we propose a four-layered architecture and corresponding four-phased methodology to enable organizations to (1) define ecological characteristics, (2) sense and measure these ecological characteristics, (3) identify, localize and visualize their environmental impact, and (4) help them to develop appropriate adaptation strategies in order to optimize their environmental impact without neglecting the organization's competitiveness.

2.9.2 Background

Organizations are based on their underlying business processes. When using service compositions to implement those business processes, certain Quality of Services need to be considered in order to achieve a successful process execution considering several internal and external constraints and objectives. Key Ecological Indicators are specific Key Performance Indicators and strategically define the organizations objectives with respect to their environmental impact. Thus, organizations need adequate methodologies and techniques to properly design their service compositions based on their environmental requirements.

2.9.3 Problem Statement

The general problem is the identification of environmentally-bad parts of business processes in terms of choosing a "good" service composition that changes the process into a more sustainable one. However,

current approaches do not cover specific environmental optimization objectives to optimize corresponding service compositions and business processes. Organizations need methods and technologies to define, measure, and analyze their business processes with respect to their economic and environmental impact. A solution to this problem is relevant in order to cope with increasing awareness of customers and legislative requirements.

2.9.4 Contribution Relevance

The paper describes an architecture and a corresponding methodology that describes one way how organizations can define, measure, analyze and adapt service compositions. The approach extends existing management approaches by introducing an environmental dimension.

2.9.5 Contribution Summary

The contribution of this paper is twofold: Firstly, we introduce an architecture that includes four layers to serve the different aspects of gBPR: (1) Strategy, (2) Sensing & Monitoring, (3) Analysis & Management, and (4) Adaptation. This architecture covers the proper monitoring, analysis and adaptation of green reengineering approaches and thus helps organizations to identify the relevant aspects for optimizing their environmental impact. Secondly, we introduce a methodology to enable the process stakeholders to reduce the environmental impact utilizing the proposed architecture.

2.9.6 Contribution Evaluation

A prototype for simulation is still ongoing work and not included in the paper. However, we used an exemplary use case in order to describe the feasibility of the approach.

2.9.7 Relation to the Research Framework

The conceptual approach proposed in this work consists of different cross-cutting concerns and is related to the Business Process Management and Service Composition domain layers within the conceptual research framework of S-Cube. As the focus is on how to optimize service compositions that build the business processes of an organization, it covers the “Engineering and Design” and “Adaptation and Monitoring” perspectives. It also covers the “Quality Definition” perspective as KEIs are a special kind of KPIs that define certain quality requirements. Based on the addressed layers and perspectives of the conceptual research framework, the approach covers the following phases of the S-Cube Reference Lifecycle: Requirements Engineering and Design, Construction, Deployment and Provisioning, as well as Operation and Management.

2.9.8 Conclusions

The architecture presented in this paper describes fundamental layers needed to achieve more sustainable organizational environments in the cross-cutting concern of green Business Process Reengineering. We described each layer in detail, identified the roles within an organization responsible for each layer and sketched the main issues of each layer. Moreover, the corresponding methodology presented in this work describes a walk through this architecture. It helps organizations to plan and define their ecological objectives in form of Key Ecological Indicators (KEIs) and to identify and localize the most dissipative parts of their processes based on these KEIs. To realize the Analysis & Management as centerpiece layer of the architecture we used the approach of “process views” that enables a proper visualization of the process model utilizing so-called view transformations. Consequently, in the Adaptation layer organizations can derive adaptation strategies to optimize their collective environmental impact while considering both, their ecological and economic objectives.

The approach bridges the gap of missing interconnection between existing Green IT and Green IS approaches towards a holistic environmental impact analysis and optimization in organizational structures. In our future work we will investigate a classification for KEIs and their application in intra-organizational and cross-partner environments. Within this work we will also address the problem of how to sense and monitor the environmental influence factors on a per task basis. We will further develop different process view patterns that allow organizations the application of process views in a re-usable fashion and will devise algorithms that support the trade-off between KPIs and KEIs. Discusses the key elements of the.

2.10 Tweetflows - Flexible Workflows with Twitter

Publication reference: [46] attached on page 219.

Partners: TUW

Status: Journal Paper submitted to SOCA journal (currently under revision)

2.10.1 Background

Tailored implementations of the SOA stack for mobile devices allow users to consume remote Web services. While as an implementation technology for mobile SOA, adopting the already broadly supported Web services (WS) is justified, studies show that the SOA stack is too complex to be implemented for mobile devices. In fact, the perceived complexity of the SOA stack hinders a wide spread adoption of SOA on mobile devices. Thus, we approach the subject of mobile SOA from a different perspective. First of all, we study the applicability of existing infrastructures like App Stores and mobile Apps for SOA principles. The second key aspect of our work is our focus on using social ties of mobile Service providers (users) for Service coordination purposes. Combined with human provided Services we lay the foundation for mobile SOA that is enriched with social aspects.

2.10.2 Problem Statement

In our work, we address three specific issues of mobile SOA: (1) the mapping of the SOA infrastructure (e.g. Service registry) onto the corresponding counterpart in the mobile domain (e.g., App Store), (2) a lightweight coordination language that satisfies the requirements of mobile Service coordination in a crowd setting and (3) the use of human provided Services to facilitate the integration of humans into mobile SOA.

2.10.3 Contribution Relevance

In our work, we introduced a conceptual mapping of SOA principles to infrastructure that is used in the mobile domain. Based on these principles, we showed how to use a lightweight coordination language (Tweetflows) to coordinate the execution of Services on mobile devices. The discovery and execution of mobile Services can be mediated by the user of the mobile phone. In doing so, we tap into the social network of the user.

2.10.4 Contribution Summary

Tweetflows introduce a new format and semantics for short Twitter text messages on top of its basic publish-subscribe message exchange pattern. In the style similar to the Twitter's established "re-tweet" (RT) primitive, users can use other primitives to announce, request, reply, delegate and query state of a

logical service, which can be human-provided, or automatically processed by a software that reads the messages from a Twitter stream. The mechanism uses hash-tags to identify the service and its attributes, while (due to the text message size limitation) service argument/result passing is done using (short) URLs that refer to them.

Tweetflow messages support the entire service life-cycle, including publishing, consumption, execution, and monitoring. Service requests are tweeted to find the adequate service providers. The originator (i.e., the coordinator) of the service request first posts a Tweet that contains a concatenation of actions that need to be executed sequentially (in a pipeline), where each service passes its result to the next one. The control over the execution is distributed: upon completion of a service in the pipeline, the provider is responsible for tweeting the invocation of the next service. Since Tweetflows are created *ad hoc*, their structure can be dynamically adjusted. Finally, an automatic binding between Tweetflows and WSDL/SOAP descriptions of human-provided services (HPS) can be generated (semi-)automatically.

2.10.5 Contribution Evaluation

We evaluated our approach by implementing several prototypes for the iOS and Android platform.

2.10.6 Relation to the Research Framework

This contribution is related to the JRA-2.2 research challenge *QoS Aware Adaptation of Service Compositions*, where the QoS aspect is mostly related to the capability and availability of human participants to perform the required service, but may also reflect the dynamic nature of the crowdsourcing environment in which it runs.

2.10.7 Conclusions

We have presented an approach for the application of SOA principles on mobile Apps. Specifically, we have investigated the mapping of existing infrastructure (Apps, App Store) to SOA concepts like Service, Service registry and Service consumer. After establishing the mapping, we have introduced a lightweight communication schema (Tweetflows) that uses social network structures and provides for the integration of human-provided services. Our next steps will be the extension of the communication schema to support more complex processes and the investigation of the distributed execution of Tweetflows in crowd scenarios. Another area of future work is the implementation of intelligent message forwarding mechanisms for the crowd. If we want to move our approach beyond the personal character of mobile Service, we need to consider the crowd as a whole as a Service provider network. If a message is sent to the crowd, the message needs to be automatically forwarded to other users, without any need for user intervention. To increase the probability of reaching the intended service, we require a directed message forwarding, based on heuristics with Twitter user profile data.

2.11 A Penalty-based Approach for QoS Dissatisfaction using Fuzzy Rules

Publication reference: [35] attached on page 226.

Partners: UCBL, UPD, PolMi

Status: Published in at ICSOC 2011, December 2011.

2.11.1 Background

While selecting services, the Quality of Service (QoS) guarantees are commonly defined in Service Level Agreements (SLAs) between provider and consumer of services. Such guarantees are often violated due to various reasons. QoS violation requires a service adaptation and penalties have to be associ- ated when promises are not met. However, there is a lack of research in defining and assessing penalties according to the degree of violation. The need is to applying penalties for QoS violations when partial satisfaction occurs.

2.11.2 Problem Statement

The contribution addresses the problem of applying penalties for QoS violations when partial satisfaction occurs. In fact, QoS guarantees defined in contracts may be violated due to various reasons. This situation needs to be handled through applying adaptation techniques not to bring dissatisfaction. The concept of penalty has been used in SLAs to compensate the conditions under which guarantee terms are not met. Despite some research have been done on the description, negotiation and monitoring of SLAs, however there is not much work on the definition of penalty clauses. studied on WS-Agreement specification to define penalties based on different types of violation. However, penalties are assigned to violation of a single property instead of assigning penalties to violation of overall QoS. Moreover, the approach introduces a method for measuring penalties which is for a predefined number of violations, instead of measuring the extent of violation and assigning penalties accordingly. One main issue is how to determine the appropriate amount of penalties as compensations from providers to satisfy customers. As quality parameters can be satisfied partially, the assessment of penalties can be based on the degree of quality violation. Understanding the violation degree is a prerequisite for assessing penalties.

2.11.3 Contribution Relevance

As quality parameters can be satisfied partially, the assessment of penalties can be based on the degree of quality violation. Understanding the violation degree is a prerequisite for assessing penalties. However, measuring such violation is yet an open research challenge. In addition, the influencing factors in defining penalties need to be identified. A static amount of penalty (manual approaches) does not re ect the extent of violation at runtime. The amount and level of penalties are related to the degree of quality violation provided from the provider side. On the other side, the customers characteristics may also affect the amount of penalties. For example a penalty to satisfy a gold/loyal customer is dierent with the one for an occasional customer. To the best of our knowledge, there is no formal relation between the assigned penalty and its influencing factors. The authors are describing the relation my means fuzzy logic.

2.11.4 Contribution Summary

The goal of this work is to apply an inference technique using fuzzy logic as a solution to propose a penalty-based approach for compensating conditions in which quality guarantees are not respected. Fuzzy logic is well suited for describing QoS and measuring quality parameters. We demonstrate a penalty inference model with a rule based mechanism applying fuzzy set theory. Measuring an appropriate value for penalties with respect to the amount of violation is the main contribution of the paper.

2.11.5 Contribution Evaluation

We have simulated our approach in a simulator based on fuzzy inference system. Initial membership functions were designed based on the contract in the motivating example and fuzzy rules are de

ned by the expert of the system. Figure 2c illustrates membership function for QoS violation. Having defined the QoS violation, we measure the extent of penalties taken into account the state of customers and previously applied penalties for the same service. For this, fuzzy rules are defined considering

all three influencing factors. Some figures depicts fuzzy rules for penalty based on QoS violations, customer's state and service status with respect to previous penalties which are de

ned by the service-state parameter represented by fuzzy set

2.11.6 Relation to the Research Framework

This contribution is directly related to the WP-JRA-2.2 research challenge *Analysis and Prediction of Quality Characteristics of Service Compositions*, and is also related to the research challenge *QoS Aware Adaptation of Service Compositions*.

2.11.7 Conclusions

Applying penalties is a complex research issue in service oriented computing which has not been paid enough attention in the literature. In this work, we elaborated the concept of penalty and propose a mechanism for modelling and measuring penalties. Penalties are modelled using a fuzzy approach and applying fuzzy set theory. The relation between penalties and their influencing factor are defined by fuzzy rules through an inference method. We have demonstrated the proposed penalty model through a motivating example and performed some initial result in measuring penalties.

Chapter 3

Conclusions

3.1 Summary

The focus of this deliverable is to present mechanisms for QoS-aware, coordinated service compositions. The contributions presented in it address that problem from several perspectives, and using different approaches, including:

- Detecting defects in realizability of service choreographies and providing diagnostics at design-time;
- Inferring user-defined attributes of orchestration activities and complex control constructs based on the notion of data sharing and an automated static analysis;
- Deriving logical specifications for service compositions from component specifications;
- Using aspect-based fragment substitution to prevent SLA violations based on machine learning predictions;
- Deriving adaptation strategies based on prediction of KPI violations in service compositions;
- Automatic composition of service fragments based on contextual requirements;
- Predicting future contexts for service compositions based on a learning model;
- Providing a framework for context-aware business process evolution;
- Proposing an architecture and a methodology for a green business process reengineering;
- Using Twitter to organize and execute (human provided) service workflows in mobile, distributed, and dynamic project teams; and
- Inferring the amount of penalties that the providers need to compensate to the service users in cases of SLA violations.

The results presented in this deliverable address the following research challenges in WP-JRA-2.2:

- Formal Models and Languages for QoS-Aware Service Compositions;
- Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies;
- Analysis and Prediction of Quality Characteristics of Service Compositions; and
- QoS Aware Adaptation of Service Compositions.

Several contributions are related to other work-packages, most notably WP-JRA-1.3 and WP-JRA-1.2, and address their corresponding research challenges.

3.2 An Outline of Future Work

The work presented in this deliverable extends the previous work within this workpackage on algorithms and techniques for splitting and merging service compositions (deliverable CD-JRA-2.2.3), on models, mechanisms and protocols for coordinated service compositions (deliverable CD-JRA-2.2.4), and on derivation of QoS for services and service compositions (deliverable CD-JRA-2.2.5). While this deliverable is the last in the S-Cube series in JRA-2.2, the following directions for the outline of the future work by the partners as well as by the wider academic and industrial community have been identified:

Bibliography

[1] George Baryannis, Manuel Carro, and Dimitris Plexousakis. Towards deriving specifications for composite web services. Submitted for publication.

[2] Piergiorgio Bertoli, Raman Kazhamiakin, Massimo Paolucci, Marco Pistore, Heorhi Raik, and Matthias Wagner. Control flow requirements for automated service composition. In *Proc. ICWS 2009*, pages 17–24, 2009.

[3] 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.

[4] Antonio Bucchiarone, Annapaola Marconi, Marco Pistore, and Heorhi Raik. CAptEvo: Context-aware Adaptation and Evolution of Business Processes. In *ICSOC 2011*, 2011.

[5] Tevfik Bultan, Xiang Fu, and Jianwen Su. Analyzing conversations: Realizability, synchronizability, and verification. In Luciano Baresi and Elisabetta Di Nitto, editors, *Test and Analysis of Web Services*, pages 57–85. Springer, 2007.

[6] Guanling Chen and David Kotz. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Dartmouth College, 2000.

[7] O. Danylevych. S-Cube deliverable CD-JRA-2.2.3: Algorithms and techniques for splitting and merging service compositions. Technical report, S-Cube Consortium, 2009.

[8] Gero Decker, Oliver Kopp, Frank Leymann, and Mathias Weske. Bpel4chor: Extending bpel for modeling choreographies. In *ICWS*, pages 296–303. IEEE Computer Society, 2007.

[9] A. K. Dey and G. D. Abowd. Towards a Better Understanding of Context and Context-Awareness. Technical report, Georgia Institute of Technology, 1999.

[10] Hanna Eberle, Tobias Unger, and Frank Leymann. Process fragments. In *OTM Conferences (1)*, pages 398–405, 2009.

[11] Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.

[12] D. Ivanović. S-Cube Deliverable CD-JRA-2.2.5: Derivation of QoS and SLA specifications. Technical report, S-Cube Consortium, 2011.

[13] D. Ivanović, M. Carro, and M. Hermenegildo. Automated attribute inference in complex service workflows based on sharing analysis. In *Proceedings of the 8th IEEE Conference on Services Computing SCC 2011*. IEEE Press, July 2011.

[14] Dragan Ivanović, Manuel Carro, and Manuel Hermenegildo. Automatic Fragment Identification in Workflows Based on Sharing Analysis. In Mathias Weske, Jian Yang, Paul Maglio, and Marcelo Fantinato, editors, *Service-Oriented Computing – ICSOC 2010*, number 6470 in LNCS. Springer Verlag, 2010.

[15] Anthony Jameson and Frank Wittig. Leveraging Data About Users in General in the Learning of Individual User Models. In *Proc. of. 17th Int. Joint Conf. on Artificial Intelligence*, pages 1185–1192. Morgan Kaufmann, 2001.

[16] Nickolas Kavantzas, David Burdett, Gregory Ritzinger, Tony Fletcher, Yves Lafon, and Charlton Barreto. Web services choreography description language version 1.0. Technical Report ws-cdl-10, W3C, 2005.

[17] Raman Kazhamiakin and Marco Pistore. Analysis of realizability conditions for web service choreographies. In Elie Najm, Jean-François Pradat-Peyre, and Véronique Donzeau-Gouge, editors, *FORTE*, volume 4229 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2006.

[18] Raman Kazhamiakin, Branimir Wetzstein, Dimka Karastoyanova, Marco Pistore, and Frank Leymann. Adaptation of Service-Based Applications Based on Process Quality Factor Analysis. In *Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+)*, 2009.

[19] Jochen Malte Küster, Ksenia Ryndina, and Harald Gall. Generation of business process models for object life cycle compliance. In *Proc. BPM'07*, pages 165–181, 2007.

[20] Philipp Leitner, Anton Michlmayr, Florian Rosenberg, and Schahram Dustdar. Monitoring, Prediction and Prevention of SLA Violations in Composite Services. In *Proceedings of the 2010 IEEE International Conference on Web Services (ICWS'10)*, 2010.

[21] Philipp Leitner, Branimir Wetzstein, Dimka Karastoyanova, Waldemar Hummer, Schahram Dustdar, and Frank Leymann. Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution. In *Proceedings of the 8th International Conference on Service Oriented Computing (ICSOC 2010)*. Springer Berlin Heidelberg, December 2010.

[22] Philipp Leitner, Branimir Wetzstein, Florian Rosenberg, Anton Michlmayr, Schahram Dustdar, and Frank Leymann. Runtime Prediction of Service Level Agreement Violations for Composite Services. In *Proceedings of the 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC'09)*, 2009.

[23] Chen Li, Manfred Reichert, and Andreas Wombacher. Discovering reference models by mining process variants using a heuristic approach. In *Proc. BPM'09*, pages 344–362, 2009.

[24] J.W. Lloyd. *Foundations of Logic Programming*. Springer, 2nd Ext. Ed., 1987.

[25] Niels Lohmann and Karsten Wolf. Realizability is controllability. In Cosimo Laneve and Jianwen Su, editors, *WS-FM*, volume 6194 of *Lecture Notes in Computer Science*, pages 110–127. Springer, 2009.

[26] Niels Lohmann and Karsten Wolf. Artifact-centric choreographies. In *ICSOC*, volume 6470, pages 32–46, 2010.

[27] Michele Mancioppi and Olha Danylevych. Awareness-based realizability analysis of choreographies. Technical Report 2012/01, IAAS, University of Stuttgart, 2011.

[28] Michele Mancioppi, Mikhail Perepletchikov, Caspar Ryan, Willem-Jan van den Heuvel, and Mike P. Papazoglou. Towards a quality model for choreography. In Asit Dan, Frederic Gittler, and Farouk Toumani, editors, *ICSOC/ServiceWave Workshops*, volume 6275 of *Lecture Notes in Computer Science*, pages 435–444, 2009.

[29] R. Mayrhofer. *An Architecture for Context Prediction*. PhD thesis, Johannes Kepler University Linz, 2004.

[30] R. Mayrhofer. Context Prediction based on Context Histories: Expected Benefits, Issues and Current State-of-the-Art. In *Proc. of ECHISE '05*, pages 31–36, 2005.

[31] K. Muthukumar and M. Hermenegildo. Compile-time derivation of variable dependency using abstract interpretation. *Journal of Logic Programming*, 13(2/3):315–347, July 1992.

[32] Elisabetta Di Nitto, Valentina Mazza, and Andrea Mocci. CD-IA-2.2.2: Collection of industrial best practices, scenarios and business cases. Technical report, S-Cube Network of Excellence, May 2009.

[33] Alexander Nowak, Frank Leymann, David Schumm, and Branimir Wetzstein. An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering. In *Proceedings of the 1st International Conference on ICT as Key Technology for the Fight against Global Warming - ICT-GLOW 2011*, volume 6868 of *Lecture Notes in Computer Science*, pages 150–164. Springer, August 2011.

[34] OMG. Business Process Model and Notation Version 2.0. OMG Specification formal/2011-01-03, Object Management Group, January 2011.

[35] Barbara Pernici, Seyed Hossein Siadat, Salima Benbernou, and Mourad Ouziri. A penalty-based approach for QoS dissatisfaction using fuzzy rules. In *ICSOC*, pages 574–581, 2011.

[36] Jan Petzold. *State Predictors for Context Prediction in Ubiquitous Systems*. PhD thesis, University of Augsburg, 2005. (In German).

[37] M. Pistore, P. Traverso, P. Bertoli, and A. Marconi. Automated synthesis of composite bpel4ws web services. In *Proceedings of the IEEE International Conference on Web Services*, ICWS '05, pages 293–301, 2005.

[38] 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.

[39] Gwen Salaün and Tevfik Bultan. Realizability of choreographies using process algebra encodings. In Michael Leuschel and Heike Wehrheim, editors, *IFM*, volume 5423 of *Lecture Notes in Computer Science*, pages 167–182. Springer, 2009.

[40] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. In *Proceedings of PODC '96*, pages 1–7. ACM, 1996.

[41] S. Sigg. *Development of a Novel Context Prediction Algorithm and Analysis of Context Prediction Schemes*. PhD thesis, University of Kassel, 2008.

[42] Adina Sirbu, Annapaola Marconi, Marco Pistore, Hanna Eberle, Frank Leymann, and Tobias Unger. Dynamic composition of pervasive process fragments. In *ICWS*, pages 73–80, 2011.

[43] Douglas R. Smith. Derived preconditions and their use in program synthesis. In Donald W. Loveland, editor, *CADE*, volume 138 of *Lecture Notes in Computer Science*, pages 172–193. Springer, 1982.

[44] Jianwen Su, Tevfik Bultan, Xiang Fu, and Xiangpeng Zhao. Towards a theory of web service choreographies. In Marlon Dumas and Reiko Heckel, editors, *WS-FM*, volume 4937 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2007.

[45] M. Treiber. S-Cube Deliverable CD-JRA-2.2.4: Models, mechanisms and protocols for coordinated service compositions – final version. Technical report, S-Cube Consortium, 2010.

[46] Martin Treiber, Daniel Schall, Schahram Dustdar, and Christian Scherling. Tweetflows – flexible workflows with twitter. In *Proceedings of PESOS 2011*. ACM, May 2011.

[47] Wil M. P. van der Aalst, Paulo Barthelmess, Clarence A. Ellis, and Jacques Wainer. Workflow modeling using proclets. In *Cooperative Information Systems, 7th International Conference, CoopIS 2000, Eilat, Israel, September 6-8, 2000, Proceedings*, pages 198–209, 2000.

[48] B. Wetzstein, A. Zengin, R. Khazamiakin, A. Marconi, A. Pistore, D. Krastoyanova, and F. Leymann. Preventing kpi violations in business processes based on decision tree learning and proactive runtime adaptatio. *Journal of Systems Integration*, 1(3):3–18, 2012. Available at: http://www.si-journal.org.

[49] Branimir Wetzstein, Dimka Karastoyanova, Oliver Kopp, Frank Leymann, and Daniel Zwink. Cross-organizational process monitoring based on service choreographies. In Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung, editors, *SAC*, pages 2485–2490. ACM, 2010.

[50] Branimir Wetzstein, Philipp Leitner, Florian Rosenberg, Schahram Dustdar, and Frank Leymann. Identifying Influential Factors of Business Process Performance Using Dependency Analysis. *Enterprise Information Systems*, Vol. 5(1):79–98, February 2011.

[51] Hongli Yang, Xiangpeng Zhao, Chao Cai, and Zongyan Qiu. Exploring the connection of choreography and orchestration with exception handling and finalization/compensation. In John Derrick and Jüri Vain, editors, *FORTE*, volume 4574 of *Lecture Notes in Computer Science*, pages 81–96. Springer, 2007.

[52] Johannes Maria Zaha, Alistair P. Barros, Marlon Dumas, and Arthur H. M. ter Hofstede. Let's dance: A language for service behavior modeling. In Robert Meersman and Zahir Tari, editors, *OTM Conferences (1)*, volume 4275 of *Lecture Notes in Computer Science*, pages 145–162. Springer, 2006.

[53] Sonja Zaplata, Matthias Meiners, and Winfried Lamersdorf. Designing future-context-aware dynamic applications with structured context prediction. *Software – Practice & Experience (SPE), to appear*, 2011.