Grant Agreement N° 215483

Title:	*Models, Mechanisms and Protocols for Coordinated Service Compositions – Final Version*
Authors:	*TUW, UCBL, UMAN, UniHH, UOC, UPM, USTUTT, VUA*
Editor:	*Martin Treiber (TUW)*
Reviewers:	*Harald Psaier, Philipp Leitner, Dragan Ivanovic*
Identifier:	*CD-JRA-2.2.4*
Type:	*Deliverable*
Version:	*1.2*
Date:	*17 September 2010*
Status:	*Final*
Class:	*External*

Management Summary

This deliverable describes models, mechanisms and protocols for coordinated service compositions. We present research results in some areas of this framework, in particular on models of service compositions, monitoring and adaptation of service compositions. The deliverable positions the individual contributions with regard to the reference lifecycle model. The work will be continued and extended in the follow-up deliverables.

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/

Contents

1 Introduction

Service compositions are a central part of of Service Based Applications (SBAs) [23]. The main benefit of Service compositions it to allow the creation of novel services from already existing ones and thus creating ”value” for the Service providers [40]. However, service compositions are nothing static, they must constantly cope with continuously evolving requirements, environments, business goals and strategies. As a consequence, the ability to monitor those aspects and adapt Service compositions accordingly is one key aspect of SBAs [10].

In this deliverable, we extend the work that we presented in the prequel deliverable CD-JRA-2.2.2 [34]. There, we presented contributions on the WP research challenges (1) Formal Models and Languages for QoS-Aware Service Compositions, (2) Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies, and (3) QoS Aware Adaptation of Service Compositions. In this deliverable, we further contribute to those challenges, partly extending previuos work and also adding new approaches. For each contribution we provide a summary, its relation to the S-Cube lifecycle phases, and involved roles (e.g., developer, service integrator) with respect to the work presented in deliverable [10].

The rest of deliverable is structured as follows. Section 1.1 provides an overview of the presented contributions and their relations to the WP research challenges. Section 1.2 explains how the contributions are related to the S-Cube reference lifecycle. In Section 2, we provide the summaries of the papers and conclude with final remarks in Section 3.

1.1 Overview of the Contributions and their Relations to Research Challenges

This deliverable is based on eight papers which were produced as joint research efforts of partners in the workpackage WP-JRA-2.2 in year 2. In this deliverable, all the papers present techniques and models for the coordinated execution of service compositions. Our particular focus lies on three main areas, namely (1) Service Composition Models and the corresponding research challenge ”Formal Models and Languages for QoS-Aware Service Compositions”, (2) Adaptation of Service Compositions related to the research challenge ”QoS Aware Adaptation of Service Compositions”, and (3) Management and Monitoring of Service Compositions related to the corresponding research challenge ”Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies”. In the following, we give a short summary of the papers and their relation to the WP research challenges and research questions as defined in the IRF.

- Service Composition Models:
  - ”The Frame Problem in Web Service Specifications” in Section 2.1 illustrates the effects of the frame problem in atomic and composite service models and proposes a solution schema, based on an existing solution in the field of procedural specifications. This solution is expressed in the form of two algorithms, one for atomic and one for composite services and is successfully applied to OWL-S service descriptions. This work contributes to the research challenge ”Formal Models and Languages for QoS-Aware Service Compositions” and addresses the corresponding research question ”Addressing the frame problem in service specifications”.
  - ”A Model-Driven Approach to Implementing Coordination Protocols in WS-BPEL” in Section 2.2 describes a model-driven approach to implementing service coordination models in WS-BPEL. A coordination protocol is specified as a coordination protocol graph (CPG). The CPG which is a platform independent model is then transformed to a set of business process models implementing coordinator and participants roles in the coordination protocol. This work contributes to the research challenge ”Formal Models and Languages for QoS-Aware Service Compositions”.
- Adaptation of Service Compositions:

  - ”Applying context to merge composition fragments in socially oriented service compositions” in Section 2.3 discusses how to ground splitting and merging in HPS concepts like (task) delegation and (task) coordination in workflows with regard to contextual information. This is done by defining regions in service compositions that can change during the execution (so called context channels) without affecting other parts of the composition. The paper devises a service composition model with a lightweight composition language based on Expressflow and shows how adaption works with context channels. This work contributes to the research challenge ”QoS Aware Adaptation of Service Compositions” and in particular to the research question ”Context-Aware Execution of Distributed Processes”.
  - ”Adaptation of SBAs based on Process Quality Factor Analysis” in Section 2.4 extends our previous work on dependency analysis of KPI violations in service compositions which has been described in the deliverable CD-JRA-2.2.2. We show how analysis results shown as dependency trees can be used for adapting service compositions so that future KPI violations are prevented. The adaptation approach includes extraction of adaptation requirements from the tree, modeling of adaptation actions, and identification and selection of adaptation strategies. This work contributes to the research challenge ”QoS Aware Adaptation of Service Compositions” and in particular to the research question ”Adaptation of QoS-aware Service Compositions based on Influential Factor Analysis and Prediction”.
  - ”Discovery and Selection of Composition-Fragments for re-Combination” in Section 2.5 discusses the discovery and selection of fragments with respect to a given composition goal to be achieved. To this end, we present an approach aiming at discovering fragments regarding their Description Logic based-semantic descriptions. The discovery is performed by applying different levels of semantic matchmaking between fragments and goals descriptions. On top of the discovery process, a Multi-Agent-System is introduced to select the most relevant fragments based on (i) non functional criteria such as their execution price (described by service providers), (ii) criteria related to their cohesion, overlap (inferred from the fragmentation technique) and also (iii) the quality of semantic matchmaking between fragments and goals. The selection process is based on a agents-oriented negotiation where each agent is responsible for a semantically coherent list of fragments. Finally, we present an approach which minimizes the number of fragments relevant for the composition goal by both (i) minimizing their overlap and (ii) maximizing the number of goals they could achieve. This work contributes to the research challenge ”QoS Aware Adaptation of Service Compositions”.
  - ”Using Soft Constraints to Make QoS-Aware Service Selections” in Section 2.6 introduces a formalization of a framework which aims at using soft constraints instead of traditional constraints in order to formulate and solve service selection problems in presence of QoS and SLAs. The advantage of this approach is that overconstrained problems, for which no complete solution exists, are instead given approximate solutions which can violate some requirements. This work contributes to the research challenge ”QoS Aware Adaptation of Service Compositions”.

- Management and Monitoring of Service Compositions:

  - ”Realizing Ad-hoc Management Capabilities for Distributed Processes” in Section 2.7. Advanced business processes are mostly distributed and require highly flexible management capabilities. However, required possibilities to take influence on a remote process execution and to react to the observed behavior of the process (preferably in real time) are still not sufficiently supported by todays process management systems. This contribution proposes a two-tier concept for monitoring and controlling distributed processes in order to flexibly collect information about the execution of process parts running on a remote system, to automatically process this information and to predefine and execute timely reactions to detected complex situations where ever necessary. As a result, ad-hoc management capabilities are enabled for processes spanning several organizations even without prior modification of the functional process. This work contributes to the research challenge ”Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies” and in particular to the research question ”Process Monitoring in Service Choreographies”.

– ”Cross-Organizational Process Monitoring based on Service Choreographies” in Section 2.8 describes an approach to monitoring of service choreographies in cross-organizational scenarios. The choreography participants specify events they require and provide to others based on BPEL4Chor models in a monitoring agreement. The monitoring agreement supports both basic events needed for process tracking and complex metrics based on complex event processing rules. This work contributes to the research challenge ”Monitoring of Quality Characteristics of Service Orchestrations and Service Choreographies” and in particular to the research question ”Process Monitoring in Service Choreographies”.

1.2 Relation to the S-Cube Life Cycle

We have positioned all papers with regard to the reference life cycle model as shown in Figure 1. It shows that all seven phases of the reference life cycle are covered by our work. However, it can be observed, that the main focus is put on *Construction*, *Operation and Managment*, *Deployment and Provisioning* and *Enact Adaption* with five, four contributions respectively. In contrast, the *Identify Adaptation Strategy* phase is covered by a single contribution.

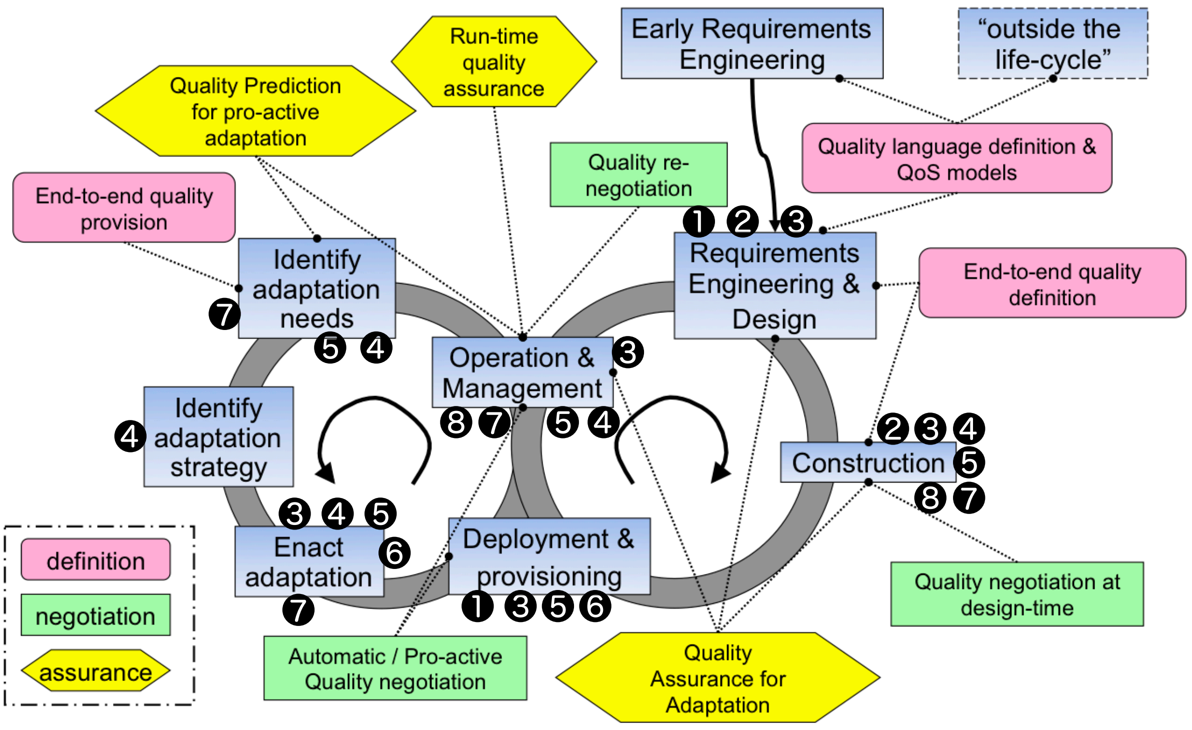
Run-time quality assurance
Early Requirements Engineering
“outside the life-cycle”
Quality Prediction for pro-active adaptation
Quality re-negotiation
Quality language definition & QoS models
End-to-end quality provision
❶ ❷ ❸
Requirements Engineering & Design
Identify adaptation needs
❼
❺ ❹
End-to-end quality definition
Operation & Management ❸
❽ ❼ ❺ ❹
Identify adaptation strategy ❹
❷ ❸ ❹
Construction ❺
❽ ❼
❸ ❹ ❺
Enact adaptation ❻
❼
Deployment & provisioning
❶ ❸ ❺ ❻
definition
negotiation
assurance
Automatic / Pro-active Quality negotiation
Quality Assurance for Adaptation
Quality negotiation at design-time

Figure 1: Reference Lifecycle Model

2 Mechanisms and Models for the coordinated execution of Service Compositions

This section summarizes the research papers that are attached to the deliverable and positions the contribution with regard to the S-Cube lifecycle model. The actual papers can be found in the respective sections of the Appendix (see also Section 1.1).

2.1 The Frame Problem in Web Service Specifications

As it has been argued in deliverable CD-JRA-2.2.2, formal specifications using the precondition/postcondition notation (as is the case for Web service specifications) are prone to a family of problems, including the frame problem. In this paper, the effects of the frame problem are explored through a motivating example of a composite service specification. In addition, a solution approach is proposed, based on knowledge gained from related research on the frame problem in procedure specifications [5]. The solution can be easily adapted and integrated in existing Semantic Web service frameworks such as OWL-S [20], WSMO [29] and SWSO [3]. For the case of OWL-S service descriptions, an algorithm is proposed, that applies the presented solution in order to transform these descriptions to ones that don't suffer from the frame problem.

2.1.1 Addressing the Frame Problem

The frame problem stems from the fact that including clauses that state only what is changed when preparing formal specifications is inadequate since it may lead to inconsistencies and compromise the capacity of formally proving certain properties of specifications. One should also include clauses, called frame axioms, that explicitly state that apart from the changes declared in the rest of the specification, nothing else changes. Solving the frame problem means expressing frame axioms without resulting in extremely lengthy, complex, possibly inconsistent, obscure specifications and at the same time retaining the ability of proving formal properties of the specifications.

The proposed solution involves declaring, for each element of the service specifications we are creating, which services may result in changing them. Thus, we don't aim to write a set of frame axioms for each individual Web service specification, but we create assertions that explain the circumstances under which each predicate or function might be modified from one state to another. These assertions, called explanation closure axioms or change axioms, provide a state-oriented perspective to specifications. To be able to express the change axioms, a simple extension to the first-order predicate logic is proposed, that adds a special predicate symbol, named Occur and a special variable symbol named α. The semantics for these two additions are simple. Variable is used to refer to services taking part in the specification. Occur(α) is a predicate of arity 1 that is true if and only if the service denoted by the variable α has executed successfully.

2.1.2 Expressing change axioms in Semantic Web Service languages

Change axioms can be expressed in all Semantic Web Service frameworks by leveraging the rule languages included or supported by them. We will briefly outline the cases of OWL-S, WSMO and SWSO. In OWL-S, the most prominent Semantic Web Service framework, rules are expressed using SWRL [11]. SWRL-FOL [27], a first-order logic extension to the SWRL syntax is more appropriate for our purposes. In SWRL-FOL, the Occur predicate can be expressed as a unary predicate which has the meaning that its argument belongs to a certain OWL class. The variable α can then be expressed as an individual variable. For an example, see the attached paper in the Appendix.

As far as WSMO is concerned, the language used for expressing rules is WSML. WSML uses similar constructs to SWRL in order to express simple logical expressions such as the change axioms of our proposed solution. Thus, we again express Occur as a unary predicate and the variable α as an individual

Phase	Support	Mechanism
Operation and Management	No	-
Deployment and Provisioning	Yes	Enriched Service Description
Enact Adaptation	No	-
Identify Adaptation Strategy	No	-
Identify Adaptation Needs	No	-
Construction	No	-
Requirements Engineering and Design	Yes	Formal Specification for SBAs

Table 1: Mapping to Adaptation Lifecycle

variable. Finally, SWSO includes SWSL as its de-facto rule language which contains two sub-languages: SWSL-FOL and SWSL-Rules. SWSL-FOL, a full first-order logic language is more than adequate for expressing change axioms.

2.1.3 An algorithm for producing change axioms

Having examined the ways of expressing a change axiom, we turn our focus on sketching an algorithm for automatically producing change axioms, given a service description using the precondition/postcondition notation. A complete set of change axioms should contain one axiom for every predicate contained in all the preconditions and postconditions stated in the description. Thus, we need to search the condition clauses to find every distinct predicate included. For each one of these predicates, our actions depend on whether a corresponding change axiom already exists. If it exists, we don't need to add a new one, since we already have expressed whether the corresponding service changes that particular predicate or not. If it doesn't exist, we need to add one that reflects the change in the value of the predicate.

For the case of composite service specifications, the algorithm needs to check each participating service specification separately to create change axioms for the predicates it contains. Also, if a change axiom already exists for a predicate, it doesn't mean that we don't need to modify it, since it may have been added in a previous step, when we were dealing with a different service. As a result, we need to do a separate search for predicates in each participating service specification, and for each predicate we find, we should check if the corresponding change axiom has already been added.

2.1.4 Mapping to Adaptation Lifecycle

With regard to the overall Adaptation Lifecycle, this work is more closely associated to the Deployment and Provisioning phase of the lifecycle and especially the concept of Service Description. Integrating this work into a Service Description model will result into complete service specifications for both atomic and composite services, with no frame problem-related issues while at the same time allowing for the formal proof of certain interesting properties such as the composability of two given services or the ability to substitute one service with another.

This work is also loosely related to the Requirements Engineering and Design phase of the S-Cube lifecycle, since it is during that phase that the requirements are collected and formulated into specifications which will then need to be satisfied by a Service Based Application. The specification for that SBA should take into account the work presented here.

2.1.5 Supported Roles

This work is closely related to the *composition designer* role. The composition designer is tasked with the creation of a composition schema, among other things. In order to create a formal specification for that composition schema, the designer must take into account the effects of the frame problem and produce a specification that avoids them. This can be achieved by applying the solution schema that is analyzed in the paper and summarized in this subsection.

2.2 A Model-Driven Approach to Implementing Coordination Protocols in WS-BPEL

In distributed computing, coordination is used as a mechanism when multiple participants must jointly agree on the outcome of a computation. A well-known example are distributed transactions where atomic commitment protocols are used to agree on the success or failure of a transaction. In the domain of Web services, coordination is addressed by WS-Coordination [1]. The WS-Coordination framework supports multiple coordination protocols and is extensible.

To facilitate the realization of new coordination protocols we present a model-driven approach for the generation of WS-BPEL service compositions that are used as implementation of coordination protocols (see [14] for details). We show how coordination protocols can be modeled in graph-based diagrams and how such graphs are transformed into abstract WS-BPEL process models representing the behavior of the coordinator and the participants in the protocol.

2.2.1 Overview of the Approach

WS-Coordination defines an extensible framework for coordinating the outcome of a set of Web services contributing to a distributed computation using a coordinator and a set of participants which interact according to a coordination protocol. Coordination protocols describe the messages exchanged between the coordinator and the participants. Two types of protocols (aka coordination types) have already been defined to cover atomic transactions (WS-AtomicTransaction) and long-running business transactions (WS-BusinessActivity). However, the use of WS-Coordination is not restricted to transaction processing systems only.

Coordination protocols can be quite complex. The coordinator has to deal with a variable number of participants. The participants can be in different states at the same time and the protocol can contain many states and state transitions. The implementation of a coordination protocol is thus difficult and error-prone. To simplify and accelerate the implementation, and eliminate errors, we propose a model-driven approach.

The protocol is first modeled as a state-based coordination protocol graph (CPG). A CPG captures the different states and state changes based on the messages exchanged between coordinator and participant. In MDA terms a coordination protocol graph specifies a Platform Independent Model (PIM). The CPG is independent of any hardware or programming platform. As the platform (in MDA terms) we have decided to use WS-BPEL. This is because coordination protocols have similar needs as business processes: they define a sequence of steps and messages to be exchanged between participants in a coordinated interaction, timing issues, and how exceptional situations must be tackled.

The CPG model is automatically transformed to abstract BPEL processes (platform speficic models (PSM)) for both the coordinator and the participant roles in the coordination protocol. These BPEL process models capture the essential parts of the message exchanges between the parties and the resulting protocol state changes. The generated code reduces the need for tedious and error-prone programming concerning the communication between the coordinator and participants in the protocol. Additional protocol logic, which cannot be captured in the CPG, has to be manually added by the programmer.

2.2.2 Mapping to SBA Lifecycle

With regard to the overall Adaptation Lifecycle, this work focuses mainly on the focuses mainly on the design and construction phase (cp. Table 2). Model driven

Phase	Support	Mechanism
Requirements Engineering and Design	Yes	Design of the CPG
Construction	Yes	Automated generation of BPEL-based protocol implementation
Deployment and Provisioning	No	-
Operation and Management	No	-
Identify Adaptation Needs	No	-
Identify Adaptation Strategy	No	-
Enact Adaptation	No	-

Table 2: Mapping to SBA Lifecycle

2.2.3 Supported Roles

In the presented approach, the *service developer* designs the coordination protocol graph and completes the generated abstract BPEL processes to executable implementations.

2.3 Applying context to merge composition fragments in socially oriented service compositions

Recently, online social networks experienced tremendous success. Platforms like Facebook, Linked in, Xing to name a few of them, experienced exponential growth rates in the last years. This success stories illustrate that humans are willing to connect on a large scale and to share information and play games on such platforms in a collective manner (e.g., Mafia Wars on Facebook). In particular, the Facebook platform provides a large set of so called Facebook applications that integrate external services from other platforms like Twitter or Wordpress, thus being essentially a platform to create (albeit limited) mashups.

From a conceptual point of view, the question of leveraging these social networks for the benefit of service compositions is still largely unanswered. Prior work [35] argues that the integration of human provided services (HPS) [30] and software services in socially oriented service compositions can utilize social networks to create flexible and context aware service compositions. This work introduces integration concepts for HPS with the help of context channels that encapsulate context information of HPS in service compositions.

Applied to splitting and merging, we can derive mappings that originate from the use of HPS in service compositions. Basically, we are able to ground splitting and merging in HPS concepts like (task) delegation and (task) coordination. We discuss this in greater detail in the following section where we introduce an illustrative example. We then summarize the technical approach for the integration of HPS and software services presented in [35].

2.3.1 Subtask delegation - Creating a Project Deliverable

Consider the example of creating a deliverable for a project. Viewed from a service composition perspective, this task can be regarded as sequence of modifications of a document until that take place until a pre defined deadline. Or, in other words, a sequence of (human provided) services that is executed as service composition. During the execution of the deliverable service composition, situations may arise that require a modification of the composition. For instance, project team members might leave, deadlines can change, even the scope of the project deliverable might change. This situations are difficult to foresee and to model in existing service compositions, due to their unpredictability. Humans, on the other hand, are able to cope with this situations: they reorganize their work and adapt to new situations.

In the example we can observe different situations, where splits and merges of HPS occur. An example considers a team member which can appoint one or more substitutes that take over the responsibilities, or in other words: the actual execution of the HPS is delegated to other HPS. This activity effectively splits a HPS onto several different other HPS. The complementary activity of coordinating the delegates and combining the results can be regarded as HPS merging. In the example, after delegating the creation of a particular section of the project deliverable to different HPS, the delegator needs to merge the results of the HPS in order to generate

2.3.2 Context channels for Splitting and Merging of HPS

On a technical level, the application of splitting and merging of HPS in socially oriented service compositions is supported by context channels. Conceptually, context channels define regions in service compositions that can change during the execution of the composition without affecting other parts of the composition. The main function of context channels encapsulate context information that is required to successfully split and merge HPS in compositions. Context information such as time and location is provided by context sensors that continuously feed context channels with context information.

Applied to the deliverable example from above, a context channel might contain parallel HPS invocations where each HPS provides a section for the deliverable. Depending on context information like location or time, a HPS can delegate, i.e., split, the creation of the section to a other HPS when the deadline is near and a single HPS would not be able to finish the task on time. Note that the availability of delegate HPS can also depend on the location: some HPS might not be available on certain locations (e.g., a HPS which location context is ”Vienna” might not be able to delegate the work because of the unavailability of delegate HPS). The merge of the results is done by the delegator HPS who collects the results of the delegates. Again, depending on the context (e.g., approaching deadline), the delegator HPS might for example decide to collect only a subset of the delegate HPS results (e.g., three instead of four deliverable contributions) and modify the result (deliverable) accordingly.

2.3.3 Prototype

We've implemented a prototype based on the Genesis infrastructure [12]. We use Groovy scripts to represent service compositions and the Genesis infrastructure provides for the API to define context channels. The code snippet below shows how we represent context channels using Expressflow [37] that includes the location context and QoS context which can be accessed in the context channel to trigger adaptations.

```
<Process efid="5c21c032-091f-45a0-aaf4 ..."
name="Service␣Mashup␣Demo"
type="Service␣Mashup" ... >
<Parallel name="Parallel1" type="Activity" ... >
  <ParallelBranch>
    <Context name="Context␣Channel1"
     type="ContextChannel"
     location="Vienna" time="Today">
    ...
     </Context>
  </ParallelBranch>
  <ParallelBranch>
    <Context name="Context␣Channel2"
     type="ContextChannel"
     delegation="No" availability="100">
     ...
    </Context>
  </ParallelBranch>
</Parallel>
</Process>
```

Listing 1: Definition of Mashup Comprising Two Parallel Context Channels

2.3.4 Mapping to Adaptation Lifecycle

With regard to the S-Cube Adaptation Lifecycle, this work is closely associated to the Operation and Management, the realization of adaptations and the construction of Service compositions in the lifecycle as shown in Table 3. Our approach focuses on the runtime adaptation of Service compositions and provides mechanisms that enable the modification of Service compositions during the execution of the Service composition. We structure service compositions with context channels which are implemented as Groovy Scrips. Scripting provides us with the ability for runtime modifications while context channels serve as pre defined regions of flexibility. This allows us to create dedicated regions in Service compositions which can be modified during the execution of the Service composition. However, our approach requires the definition of context channels during the design phase to be able to adapt the Service composition. Our prototype is build on top of the Genesis framework, which offers a rich set of features (e.g., exchange of composition scripts) during the execution of Service compositions and provides us with the features to enact Service adaptations.

Phase	Support	Mechanism
Operation and Management	Yes	Genesis API
Deployment and Provisioning	Yes	Genesis API
Enact Adaptation	Yes	Genesis API, Context Channels
Identify Adaptation Strategy	No	-
Identify Adaptation Needs	No	-
Construction	Yes	Context Channels
Requirements Engineering and Design	Yes	Context Channels

Table 3: Mapping to Adaptation Lifecycle

2.3.5 Supported Roles

Our approach focuses mostly on the perspective of the developer and the service integrator, since we discuss adaptation on a technical level in our contribution. However, we also provide a more abstract perspective on Service compositions with context channels. The design of a Service composition can help designers to create Service compositions on a more abstract level, without the requirement of knowing all Service composition details. Thus, the designer can create a Service composition similar to guidelines. By specifying the mandatory parts of a Service composition and leaving parts open, the designer can create adaptive Service compositions while at the same time providing for stability regarding the overall Service composition.

In real world settings, we have observed that roles tend to overlap. The same person is the designer and Service composer at the same time. We believe, with the introduction of context channels, we provide means to create additional distinctions between the designer of a Service composition and Service composers, Service developers respectively. Context channels can help to structure Service compositions into smaller easer understandable parts and help fostering a design of Service compositions that a priori are build for adaption.

2.4 Adaptation of SBAs based on Process Quality Factor Analysis

The performance of SBAs on process level is measured in terms of key performance indicators (KPIs). If monitoring of KPIs shows that KPI targets are not met, the influential factors have to be analyzed. The KPI thereby can depend on various factors from different functional layers of an SBA. In [39] (and previous deliverable CD-JRA-2.2.2) we have described an approach to analysis of influential factors of KPIs of service compositions (a.k.a. process quality factor analysis). Thereby, based on monitoring results, a decision tree (dependency tree) is constructed which shows for which values of lower-level metrics the KPI targets are mostly violated.

In the following we present a novel adaptation approach (described in detail in [13]) which is based on the previously described process quality factor analysis. We show how the analysis results can be used to come up with an adaptation strategy leading to an SBA that satisfies KPI values. The approach includes creation of a model which associates adaptation actions to process quality metrics, extraction of adaptation requirements from the dependency tree, and identification of an optimal adaptation strategy.

2.4.1 Overview of the Approach

The goal of our approach is to adapt service compositions in order to prevent the violation of KPIs. The approach consists of several phases.

At design time quality modeling for analysis and adaptation is performed. Thereby a metrics model and an *adaptaton actions model* are created. The metrics model defines the KPIs and the lower level metrics which are to be monitored and which are needed for quality factor analysis. Based on the metrics model, the user specifies available adaptation actions (e.g., service subtitution) which can be used to adapt the factors measured by the metrics. Thereby, one has to specify the effect of the adaptation action on the metric. In particular, the model allows for defining whether an action contributes positively or negatively to a certain quality factor, i.e., whether it improves the value of a metric.

At process runtime, based on the metrics model, the monitoring of KPIs and potential influential metrics is performed. Then the metrics are analyzed in order to identify the reasons, i.e., the influential factors, which lead to the undesired values of the specified KPIs. More precisely, the approach is based machine learning techniques to construct a decision tree (a.k.a. dependency tree) which shows for which value ranges of influential metrics a KPI is satisfied or violated (see [39]).

In the next step, we extract *adaptation requirements* from the dependency tree. This is done by identifying those tree paths of application metrics (and their value ranges) that correspond to the good values of the KPI (e.g., delivery time shipment ¡ 2 days AND delivery time supplier ¡ 3 days –¿ KPI satisfied). If we can make sure that one of those paths is followed for the current process instance (in case of instance adaptation) or future process instances (in case of model adaptation), then the KPI target will be satisfied. The result of this analysis characterizes thus combinations of those factors of the application that should be improved (metrics) and how they should be improved (their values).

After identifying the adaptation requirements, the next phase aims to combine concrete adaptation actions that address the identified requirements thus creating *adaptation strategies*. This phase uses the adaptation action model, where the effect of adaptation actions on different application metrics is described. It takes into account that an adaptation action contributes positively or negatively to a certain quality factor, i.e., whether it improves the value of a metric. After identification of a set of alternative adaptation strategies, one strategy is selected based on certain criteria.

The selected adaptation strategy is used for adaptation of the process model or process instance by executing all contained adaptation actions. After adaptation, the existing KPIs and metric definitions might have to be adapted thus closing the cycle.

Phase	Support	Mechanism
Requirements Engineering and Design	No	-
Construction	Yes	Modeling of metrics and adaptation actions
Deployment and Provisioning	No	-
Operation and Management	Yes	Monitoring of KPIs
Identify Adaptation Needs	Yes	Quality factor analysis and identification of adaptation requirements
Identify Adaptation Strategy	Yes	Identification of adaptation strategies based on adaptation actions model
Enact Adaptation	Yes	Service Substitution

Table 4: Mapping to SBA Lifecycle

2.4.2 Mapping to SBA Lifecycle

The contribution focuses mainly on the adaptation phases, but the overall approach relies also on monitoring and the design phases where the monitoring and adaptation models are created (cp. Table 4).

2.4.3 Supported Roles

The presented approach has relationships to several stakeholders. The *service provider* defines the KPIs and KPI targets which are to be achieved to reach business goals. The *service developer* creates the metrics model which contatins the KPIs and a set of lower level metrics needed for quality factor analysis. The *service developer* also creates the adaptation actions model based on available adaptation mechanisms. At process runtime, the *service provider* chooses the adaptation strategy which should be enacted from the alternatives proposed by our framework.

2.5 Discovery and Selection of Composition-Fragments for re-Combination

Research on web services revolved around the development of techniques for description, discovery, selection, composition, personalization and invocation of web services. Composition of web services [6, 28, 21, 33, 18], or the process of selecting and inter-connecting services provided by different partners according to a goal to achieve, is probably the most interesting challenge spawned by Service-oriented Computing paradigm [31]. One broad category of changes applied to service compositions hinges on their fragmentation [7] i.e., creating composition- fragments (in the rest of the Section we will use the term *fragment*) that group services and some service compositions elements such as activities, control flows, data flows. Therefore according to some specific criteria, it is now possible to re-combine these fragments from different compositions in order to achieve new functionalities. For instance, elaborating a composition of *fragments containing all the activities performed by the same actor*, or compositions of *fragments with robust data flow* [17] become real using macro-level descriptions of services i.e., fragments.

Due to an increasing amount of available and ready-to-use fragments, which are results of fragmentation techniques, some issues regarding their discovery, selection, re-combination and optimization need to be addressed to achieve Macro-level based compositions i.e., compositions of high level specification of processes. In this section, we focus on discovery and selection of fragments with respect to a given composition (or re-combination) goal to be achieved. To this end fragments are described using Description Logic (DL) based-semantic descriptions. Therefore, the discovery is performed by applying different levels of semantic matchmaking between fragments and goals descriptions. On top of the discovery process, a Multi-Agent-System (MAS) is introduced to select the most relevant fragments based on (i) non functional criteria such as their execution price (described by service providers), (ii) criteria related to their cohesion, overlap (inferred from the fragmentation technique) and also (iii) the quality of semantic matchmaking between fragments and goals. The selection process is based on a agents-oriented negotiation where each agent is responsible for a semantically coherent list of fragments. On top of this approach, a minimization step of the number of relevant fragments is required for both (i) minimizing their overlap and (ii) maximizing the number of goals they could achieve.

The rest of this section is organized as follows: Section 2.5.1 defines the model we use for fragments. Section 2.5.2 sketches our ideas towards our agent-based negotiation approach by defining the object we negotiate and expect to optimize. Section 2.5.3 briefly defines the process for re-combining fragments which have been selected by our approach. Finally we conclude in Section 2.5.4.

2.5.1 Fragment Description

A fragment is mainly defined by its functional description whether available: its functional category (from an ontology of domains) defining the general goal the fragment achieves, and classic parameters (inherited from the service definition): input, output parameters, preconditions and effects. These descriptions are defined along semantic descriptions in ontologies.

In addition, any fragment is defined by attributes related to the services which are involved in, for instance not only their non functional properties [25] (*availability*, *execution time*, *execution price*) but also its degree of instantiation i.e., rate of real service bound to activities in the composition. Indeed, some parts of fragments may need to be bound to service at run time due to some context adaptation issues.

Finally, fragments inherits some properties and criteria from the fragmentation techniques [7] they are originally from. In this direction, fragments are defined along i) self-containment: reflecting whether the fragment contains all data necessary for its execution, ii) cohesion, providing a measure of how many the elements in the fragment belong together and iii) overlap: defining the degree of covering with other fragments from the same composition.

2.5.2 A Negotiation-based Approach for Discovery and Selection of Fragments

Since many fragments offering same or close functionalities can achieve a same goal, it seems important to consider a method to discover and select them given some quality criteria and preferences.

Mainly due to the large amount of fragments, an agent-based negotiation strategy is required to distribute them and easily interact with their descriptions in a scalable way. According to Figure 2, different categories of autonomous and independent agents (i.e., initiator, requesters and providers) cooperate to achieve selection and discovery of fragments in order to re-combine the relevant fragments. Besides involvement of agents, our architecture in Figure 2 emphasizes the main processes of our approach: discovery, selection and re-combination.

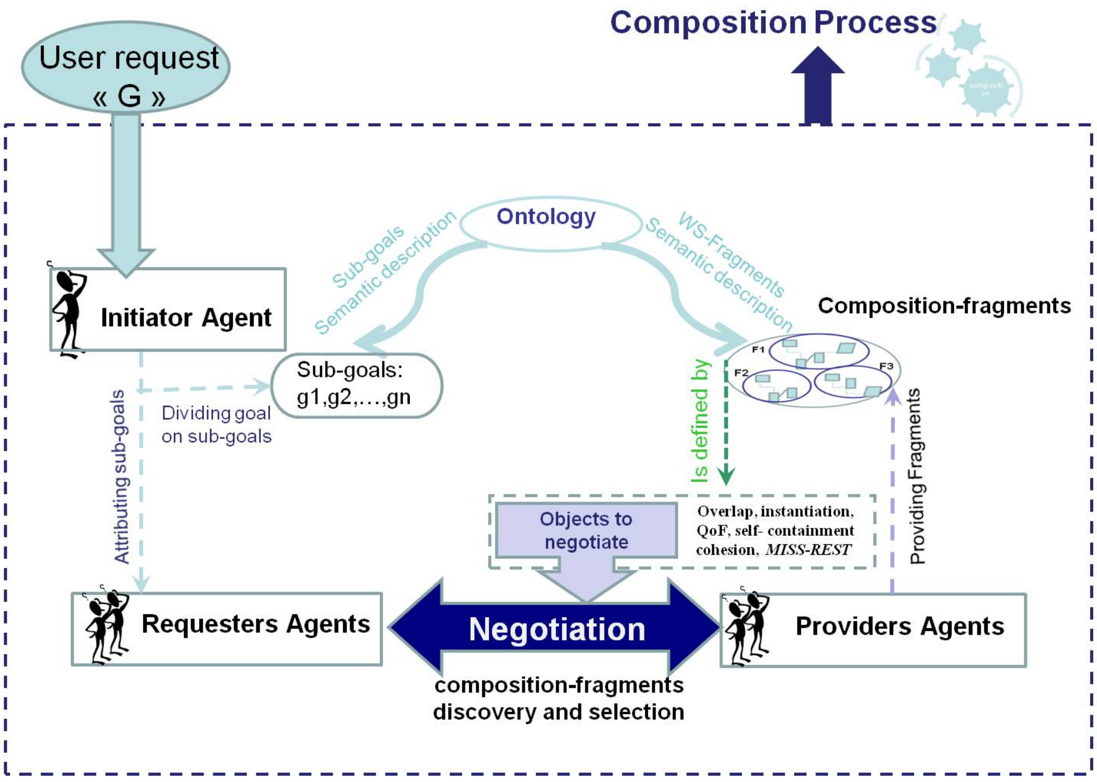

Figure 2: Discovery and Selection of Composition-Fragments for re-Combination.

The initial request is modeled as a goal G in Figure 2. Such a goal is defined as a satisfiable conjunction of DL concepts on a domain ontology. As a first step, the initiator agent is responsible for dividing the main goal as a conjunction of sub-goal $g_1 \sqcap g_2 \sqcap \ldots \sqcap g_n$. Each sub-goal g_i could be an atomic DL concept of G or even more complex by containing more than one. Once the sub-goals are described, the initiator agent allocates each sub-goal to specific requester agents. Since providers agents are responsible for a repository of fragments, negotiations act between the latter agents and the requester agents. For each requester agent, the aim of this negotiation is to obtain the most relevant fragment achieving its sub-goal by considering criteria attached to the fragment definition (see Secion 2.5.1). The final step of the process consists in re-combining selected fragments (Section 2.5.3).

Therefore, the selection step of our negotiation approach aims at maximizing the semantic matching [26] between sub-goals and fragments regarding the functional and their semantic descriptions. In addition it aims at i) minimizing the execution time and price, and ii) maximizing the availability and the instantiation of fragments (i.e., minimizing the number of unbound activities in the composition) at non functional level. Regarding the criteria derived from the fragmentation technique, the selection step aims at i) minimizing the overlap between fragments, and ii) maximizing the self-containment, the cohesion of fragments at non functional level.

Phase	Support	Mechanism
Requirements Engineering and Design	No	-
Construction	Yes	Fragment-based specification
Deployment and Provisioning	Yes	Data-Flow oriented
Operation and Management	Yes	Management Software agents and ontologies
Identify Adaptation Needs	Yes	Quality factor analysis
Identify Adaptation Strategy	No	Overall quality evaluation
Enact Adaptation	Yes	Invocation of (predefined) construction actions

Table 5: Mapping to Adaptation Lifecycle

By minimizing the overlap criterion, we minimize the number of fragments that could achieve *G*. However, this minimization approach is local rather than global. Towards this issue, we suggest a co-operative approach to obtain a global minimization by selecting the common fragments shared by more than one requester agent.

2.5.3 Automated Re-Combination of Fragments

Once fragments are discovered and selecting by optimizing their criteria, a composition process is required to re-combine these fragments and then re- organize the data flow and control flow of the new composition. To this end, we follow a state of the art functional-based approach [18] aiming at automatically composing them by exploiting their semantic and data flow description. The composition process is view, in this approach, as a composition of semantic links. The key idea is that the matchmaking enables, at run time, finding semantic compatibilities among independently defined fragment descriptions.

2.5.4 Summarization of Results

We have presented our contribution towards the discovery, selection and re-combination of fragments, emerging from many different fragmentation techniques. To this end, we presented the semantic description of fragments and introduced the most relevant criteria which are required to achieve their discovery and selection. In this work we are benefiting of the use of semantics description to provide unambiguous description of fragments, user request (or goal to be achieved), and software agents goals. From these descriptions we suggested an agent-based negotiation strategy to automatically discover and select fragments. The negotiation is based on fragments criteria and the user preferences related to these criteria.

The research area in macro-level composition of fragments is a new area, which required further work to be addressed. For instance, much work should be done on automation of discovery and selection processes of the existing fragments, for instance by enriching their descriptions, especially criteria related to their fragmentation techniques. In addition different re-combination approaches need to be compared and analyzed depending on fragments descriptions. Indeed, depending on the level of description e.g., functional [18] and process levels [28], different re-combination approaches are required.

2.5.5 Mapping to Adaptation Lifecycle

Our approach focuses on the i) design-time adaptation of the service construction and compositions, and ii) quality negotiation at design-time. In our context, compositions are combinations of fragments. Our approach provides means to re-compose macro-level descriptions of services i.e., fragments rather than atomic services. Therefore, the presented approach mainly contributes to the phase of *Construction* within the adaptation lifecycle, but has also minor effects and benefits on other phases (cp. Table 5).

2.5.6 Supported Roles

The presented approach has relationships to several stakeholders. The *service provider* defines the semantic descriptions of services and their non functional properties. The quality criteria related to the fragmentation techniques are provided during fragmentation and hosted by software agents. The *service developer* creates the metrics model which contains the quality criteria needed for quality factor analysis. The *service consumer* (who can be an outsourcing business partner or the end customer) is interested in providing preferences on final compositions which are returned by our approach.

2.6 Using Soft Constraints to Make QoS-Aware Service Selections

2.6.1 Classical Formuation Using Constraint Systems

Constraint solving [22, 8] have been used to represent and solve the problem of selecting a semantic-matching service while making sure that it fulfills the expected QoS requirements and, if several semantically valid services are available, to choose the one which optimizes some measure of quality [16], including, in some cases, semantic reasoning on the fly [15].

However complex and elaborate, a there is flaw in the ointment of reality in all of these cases: real-life problems are expected to be overwhelmingly complex, and, as such, the number (and type) of constraints imposed may very well be unsolvable. This phenomena is not exclusive of matchmaking and selection in service-based computing, but it has been observed in a plethora of real-life situations in other realms. In short, constraint-based solver which faces a contradiction (either stemming from an ill-posed problem or from an over-constrained one) will silently *fail* and return no answer whatsoever, as it is fundamentaly a *yes/no* answer system, which accompanies the *yes* answer with a witness (a valuation for the involved variables) of the positive case.

2.6.2 Workarounds for Overconstrained Problems

A possible workaround within the boolean decision case involves selecting a (maximal) subset of the initial constraint which is solvable. The drawbacks are that this is a solution of exponential complxety (added to the complexity of solving the constraints themselves) and, in principle, several such subsets may exist, which have to be generated and compared if optimality is a goal. Other approaches involve enriching the underlying constraint system. Arguably the most general setting so far is that of Soft Constraints [4].

In short, Soft Constraints can associate a quality to any solution of any constraints and define operators to keep this quality consistently when making conjuctions of constraints and when projecting on the query variables. Solutions which would be non-admissible in the traditional setting would simply have a confidence level zero. Valuations with the highest confidence level naturally correpond to optimal solutions.

2.6.3 Contribution: Extending Soft Constraints

In our paper, we propose using a variation of the aforementioned ring-based Soft Constraint framework to represent the requirements expressed in SLAs. In order to better use the power of the underlying framework, fragments of SLAs are reformulated to state explicitly preferences and the penalties to be applied in case some preference is not met. In our scheme, penalties and preferences are grouped and ranked in a specific fashion which is not straightforwardly mapped onto a standard soft constraint.

Therefore we preferred to enlarge the basic Soft Constraint formulation in order to capture natively both preferences and penalties. This enriched framework is a conservative extension of the original framework which preserves its good properties (also regarding existence of solutions) while being much more amenable to expressing complex SLAs with the possibility of not matching all the expecations. We include a sketch of architecture for SLA processing which includes constraint solving capabilities.

2.6.4 Lifecycle Mapping

Our contribution can be put to work at any time where there is a need to automatically select which services can better suit a given SLA. However, it can be used to monitor the health of a SBA by evaluating how the actual values of the variables cotribute to solving the constraint system. See Table 6.

Phase	Support	Mechanism
Requirements Engineering and Design	No	-
Construction	No	-
Deployment and Provisioning	Yes	Selection of services
Operation and Management	Partially	Initial constraints can be continuously checked during operation
Identify Adaptation Needs	No	-
Identify Adaptation Strategy	No	-
Enact Adaptation	Yes	Select optimal services

Table 6: Mapping to SBA Lifecycle

2.6.5 Supported Roles

Our approach involves the *Business Analyst* who is in charge of deciding which high-level goals have to be met, and expressing them using the SLA vocabulary, the *Service Provider* who has to describe the properties (i.e., semantics and QoS) of the available services and *Service Consumer* who has to set up well-formed constraint systems whose solutions return which service is to be selected.

2.7 Realizing Ad-hoc Management Capabilities for Distributed Processes

Advanced service compositions are mostly distributed and require highly flexible management capabilities. However, today's business process management systems mostly consider monitoring and controlling of single centralized process executions, are often heterogeneous and do not provide standardized runtime monitoring or management APIs [36]. Therefore, an integration of runtime monitoring information from different source systems is hardly possible yet. Required possibilities to also take influence on a remote process execution and to react to the observed behavior of the process (preferably in real time) are still challenging.

Therefore, this contribution aims at a concept and supporting infrastructure to flexibly collect information about the execution of process parts running on a remote system, to automatically process this information and to predefine and execute timely reactions to detected complex situations where ever necessary. As summarized in the following, the approach consists of a service-based common management interface and uses complex event processing in order to specify user-defined management rules and actions. Further details about these two components and about the technical implementation can be obtained in [42].

2.7.1 Business Process Management System as a Manageable Resource

In order to find an adequate basis for a common understanding of the elements and attributes relevant for distributed process management, an analysis of several current practical and theoretical approaches and systems as well as abstract models and concrete products for traditional and distributed business process management has been carried out. The analysis lead to the identification of most relevant management entities and a resulting basic model. It holds the process management system (including process models, process instances and relevant context information) as the manageable resource which can be accessed by a service-based management interface either by pulling read-only information about its entities (*information interface*), by asking for manipulation of entity values (*modification interface*) or by receiving events emitted by the entities (*event interface*).

According to existing approaches such as Wf-XML [32], the manageable entities contain a number of sub entities and individual atomic properties, e.g. a process engine has a current workload expressed as the number of running process instances and CPU load, or a process instance activity has a start time, a duration and an end time. Creation of entity instances and changes of their properties' values are effecting the associated *events*. In order to allow manageability, each entity is described by a uniform and unambiguous representation and interpretation of values, e.g. represented as standard or complex data types, and a metric. Furthermore, the *modifiability* (e.g. read or read-write), the *availability* of the property (e.g. before, during or after execution of the process or the activity) and the *mutability* and *frequency of updates* is specified.

Finally, the model of relevant characteristics and relationships of process management systems, process models, instances and context are represented as resource properties according to the *Web Services Distributed Management (WSDM)* standard [24]. The resource properties can be accessed using the standard WSDM web service operations *GetResourceProperty* and *UpdateResourceProperty* as well as by a set of additional operations, e.g. for cancellation of process instances, which are altogether included in the associated web service description (WSDL).

2.7.2 Specification and Processing of Management Rules

Based on the established management services and events, local or remote applications can make use of the provided functionality in order to perform monitoring or apply necessary changes on running process instances. In order to minimize delays between the detection of critical situations and the initiation of an appropriate reaction, this approach introduces an on-site management component which is able to receive user-defined *management rules* that are to be enforced during process execution. An example is monitoring the duration of executing a specific activity, and, in case a specified amount of time has passed and no progress becomes visible, to restart the activity. However, also monitoring rules that do not influence the execution of the process are possible (e.g. after each activity, its performer, duration and current location should be logged) or distribution decisions and actions can be supported (e.g. if the workload exceeds a specified threshold, the process should be transferred to a process engine with a better capacity).

Technically, a management rule consists of an *event pattern* and, optionally, an associated *management action* which is invoked when a situation (potentially consisting of a number of process-related events) applies to this event pattern. The event pattern thus represents some arbitrarily complex inference of information from one or more primitive or other complex events and is expressed in an SQL-based query language. Accordingly, management rules are processed on the basis of *Complex Event Processing (CEP)* techniques. As an example, the prototypical implementation makes use of an the existing ESPER[1] rule engine.

2.7.3 Summarization of Results

Compared to existing approaches (e.g. [2, 38]), the presented approach can handle management requirements which are known in advance (e.g. error handling strategies) as well as ad-hoc management tasks (e.g. status requests). It supports both passive retrieval of monitoring information and active manipulation of process execution. Disconnected clients can be (partly) supported by an on-site management component responsible for the execution of predefined rule-action pairs. In both cases, business logic (described by the original business process model) and management logic (described by the resource property document and management rules) are well separated.

In summary, the presented approach allows for increased flexibility during process execution – taking into account also the requirements of modern distributed process management variants such as BPM-as-a-Service or integration of mobile (sub-) systems – and the integration of valuable functionalities of remote process management systems which have not been exploited before. The price for such increased flexibility is, however, the necessity to integrate and configure a corresponding add-on infrastructure. Finally, the development of adequate security and privacy mechanisms to restrict access to private information and sensitive control options is a relevant subject to future work.

2.7.4 Mapping to Adaptation Lifecycle

The contribution focuses on runtime monitoring and ad-hoc adaptation of service compositions. It provides means in order to inspect running service compositions at runtime, thereby to detect errors or non-compliances, and finally to invoke reactions to detected situations based on predefined management rules. Therefore, the presented approach mainly contributes to the phase of *operation and management* within the adaptation lifecycle, but has also minor effects and benefits on other phases (cp. Table 7).

2.7.5 Supported Roles

The presented approach has relationships to several stakeholders. As a prerequisite, the *service provider* (i.e. the owner or operator of the process management system) has to implement the proposed Management API and allow customers to use associated management functionality. The service provider is interested in such mechanism, because offering additional management capabilities advances quality of service and thus increases the probability of being selected as a business partner. The *service consumer* (who can be an outsourcing business partner or the end customer) is interested in monitoring and controlling the execution of "his" processes in order to flexibly react to internal or external events if necessary. The provision and consumption of *parcel tracking services* is an analogous example for the relationship between these two roles.

[1] http://esper.codehaus.org/

Phase	Support	Mechanism
Requirements Engineering and Design	No	-
Construction	Yes	Specification of management rules
Deployment and Provisioning	No	-
Operation and Management	Yes	Management API and rule processing
Identify Adaptation Needs	Yes	Complex event processing
Identify Adaptation Strategy	No	(Strategy is predefined)
Enact Adaptation	Yes	Invocation of (predefined) management actions

Table 7: Mapping to Adaptation Lifecycle

Nevertheless, also technical stakeholders are involved. The *service developer* (i.e. the business process modeler) can optionally specify a set of management rules in order to automatically invoke reactions to situations which are assumed to appear during process execution. The *business analyst* uses the presented mechanism in order to collect information about running or finished process instances and, based on that, to decide about optimizations for the execution and distribution of future processes.

2.8 Cross-Organizational Process Monitoring based on Service Choreographies

Monitoring of business processes in the area of service oriented computing is typically performed using Business Activity Monitoring (BAM) technology in an intra-organizational setting. Thereby, the monitoring solution is specified based on events published by the process execution middleware as the business process (e.g. implemented as a service orchestration) is executed. Business activity monitoring has been traditionally focused on intra-enterprise processes. Today, due to outsourcing and increasing customers' needs the companies are forced to share monitoring information with each other and their customers. A well-known example is shipment tracking whereby the shipper opens its process to some extent to the customer for monitoring purposes.

In this section, we present a solution to this problem by describing an event-based approach to cross-organizational monitoring based on service choreography descriptions. The approach is described in detail in [41]. We use BPEL4Chor descriptions as basis for the specification of monitoring agreements between partners. A monitoring agreement defines which events each participant in the choreography should provide to the other participants. Thereby, we distinguish between resource events (gathered from the process engine) and complex events which are evalauted using Complex Event Processing (CEP) [19] and which are needed for metric calculation.

2.8.1 Monitoring in Service Choreographies

A service choreography description can be seen as an agreement between participants on their public processes and message exchanges. As the choreography description is an agreement between partners on their functional interfaces, we argue that it can also be used as basis for specification of (non-functional) monitorability aspects. In our approach we use BPEL4Chor [9] as a choreography language. In BPEL4Chor each participant models his public process as an abstract BPEL process. Thereby, the private parts of the process are modeled as opaque activities.

For definition of monitoring properties, we introduce a *monitoring agreement* which is an XML-based document specifying monitoring aspects between partners based on the choreography description. A monitoring agreement consists of a set of resource event definitions and complex event definitions. Resource events are defined based on abstract BPEL processes in the choreography by specifying at which BPEL instance resource (process, activity, variable) and for which state of that resource (started, completed, terminated, etc.) an event is to published, which data it should contain (e.g., which BPEL variables or parts of them to include in the event), and where it should be published (at which message queue or pub/sub topic).

```
<monitoringAgreement xmlns:chor="http://purchaseOrder/choreography"
   xmlns:reseller="http://purchaseOrder/reseller">
 <resorceEventDefinitions>
  <resourceEventDefinition name="OrderReceivedEvent">
   <monitoredResource choreography="chor:orderChoreography"
      process="reseller:ResellerProcess" scope="process"
      activity="reseller:ReceivePO" state="completed"/>
   <data>
    <processVariable name="order" variable="purchaseOrder"/>
   </data>
   <publish>
    <topic name="purchaseOrder.reseller" access="reseller,␣customer"/>
   </publish>
  </resourceEventDefinition>
   ...
 </resorceEventDefinitions>
  ...
</monitoringAgreement>
```

Listing 2: Resource Event Definition

Listing **??** shows a *resource event definition* for the `OrderReceived` resource event. It is specified by pointing to the `Receive PO` activity in the BPEL process model of the reseller in a purchase order processing choreography. The event is to be published when the corresponding activity is `completed`. In addition, the event should contain the data from the `purchaseOrder` variable. It is published to the queue which can only be accessed by the reseller.

Resource events can already be used for process tracking purposes but serve also as basis for complex events. Complex events are defined based on resource events and other complex events using a Complex Event Processing (CEP) language. In our approach we use the language from the ESPER framwork. Complex eventrs are needed for calculation of process metrics.

```
<complexEventDefinition providedBy="reseller"
  name="OrderFulfillmentTime"
  choreography="chor:orderChoreography">
 <consume>...</consume>
 <eventAggregation resultType="COMPLEX">
  <statement><![CDATA[
   SELECT
     abs(b.timestamp - a.timestamp) AS metricValue,
     "ms" AS unit,
     a.resource.ciid AS ciid
   FROM PATTERN [ EVERY
     a = ResourceEvent(
       name="OrderReceivedEvent")
     -> b = ResourceEvent(
        name="ShipmentReceivedEvent"
        AND resource.ciid = a.resource.ciid) ]
  ]]><statement>
 </eventAggregation>
 <publish>
  <topic name="orderChoreography.customer"
     access="customer,␣reseller"/>
 </publish>
</complexEventDefinition>
```

Listing 3: Complex Event Definition

In Listing 3 a complex event `OrderFulfillmentTime` is defined which contains the corresponding metric value in the attribute `metricValue`. In addition it contains the attribute `unit` and the choreography instance identifier. The metric value is calculated by correlating two events already defined, namely `OrderReceivedEvent` and `ShipmentReceivedEvent`. These events are correlated based on choreography instance IDs and then their timestamps are subtracted. The result event is published to the corresponding queue by the reseller who also performs this event aggregation. Note that obviously such a definition results in one result event per choreography instance, i.e. an event stream.

After the monitoring agreement is created, it is deployed to each participant's infrastructure. The participant thereby extracts from the agreement the events it has to provide and configures its middleware, e.g., the BPEL engine using a deployment descriptor to provide resource events, and the CEP engine to provide complex events. It also subscribes to topics or queues where he receives events from other participant.

2.8.2 Mapping to S-Cube SBA Lifecycle

The contribution focuses on runtime monitoring of service compositions. Therefore, the presented approach mainly contributes to the phase of *operation and management* within the SBA lifecycle (cp. Table 8).

Phase	Support	Mechanism
Requirements Engineering and Design	No	-
Construction	Yes	Specification of monitoring agreements
Deployment and Provisioning	No	-
Operation and Management	Yes	Complex event processing
Identify Adaptation Needs	No	-
Identify Adaptation Strategy	No	-
Enact Adaptation	No	-

Table 8: Mapping to SBA Lifecycle

2.8.3 Supported Roles

The presented approach has relationships to several stakeholders. The *service provider* has to implement mechanisms for modeling and deploying the monitoring agreement in its infrastructure and provide events as specified. The service provider is interested in providing such mechanisms, because it enhances its offering for the customers (e.g., provision of shipment tracking). The *service consumer* (who can be an outsourcing business partner or the end customer) is interested in monitoring the processes of the service provider in order to be able to flexibly and timely react to unexpected events if necessary.

3 Conclusions

This deliverable discussed new contributions in the context of models and mechanisms for coordinated QoS-aware service compositions. The work presented builds on service composition techniques discussed in CD-JRA 2.2.2 and CD-JRA 2.2.3. The approaches cover WP research challenges considering formal models, monitoring, and adaptation of service compositions. For each contribution we have provided a mapping to the reference life cycle and we have established explicit links to the work done in JRA 1.3 by discussing the stakeholders which were introduced in JRA 1.3.4.

References

[1] Web Services Coordination (WS-Coordination) Version 1.1, April 2007. `http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.1-spec-os.pdf`.

[2] Luciano Baresi, Carlo Ghezzi, and Sam Guinea. Smart Monitors for Composed Services. In *ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing*, pages 193–202, New York, NY, USA, 2004. ACM Press.

[3] Steve Battle, Abraham Bernstein, Harold Boley, Benjamin Grosof, Michael Gruninger, Richard Hull, Michael Kifer, David Martin, Sheila McIlraith, Deborah McGuinness, Jianwen Su, and Said Tabet. Semantic Web Services Framework (SWSF) Overview. World Wide Web Consortium, Member Submission SUBM-SWSF-20050909, September 2005.

[4] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint satisfaction and optimization. *J. ACM*, 44(2):201–236, 1997.

[5] Alexander Borgida, John Mylopoulos, and Raymond Reiter. On the Frame Problem in Procedure Specifications. *Software Engineering, IEEE Transactions on*, 21(10):785–798, 1995.

[6] Tevfik Bultan, Xiang Fu, Richard Hull, and Jianwen Su. Conversation specification: a new approach to design and analysis of e-service composition. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 403–410, New York, NY, USA, 2003. ACM Press.

[7] Olha Danylevych and Dimka Karastoyanova. Algorithms and techniques for splitting and merging service compositions. In *S-Cube Deliverable: CD-JRA-2.2.3*, 2009.

[8] Rina Dechter. *Constraint Processing*. Morgan Kaufman, 2003.

[9] Gero Decker, Oliver Kopp, Frank Leymann, and Mathias Weske. BPEL4Chor: Extending BPEL for Modeling Choreographies. In *ICWS*, Salt Lake City, USA, July 2007.

[10] Andreas Gehlert and Andreas Metzger. Quality reference model for SBA. Contractual Deliverable CD-JRA-1.3.2, S-Cube Consortium, March 2008. The following institutions contributed to this deliverable: City University London, Consiglio Nazionale delle Ricerche, Center for Scientific and Technological Research, The French National Institute for Research in Computer Science and Control, Lero - The Irish Software Engineering Research Centre, Politecnico di Milano, MTA SZTAKImputer and Automation Research Institute, University of Duisburg-Essen, Vienna University of Technology, Universidad Politécnica de Madrid, University of Stuttgart and Tilburg University.

[11] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosofand, and Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, May 2004. Available at: http://www.w3.org/Submission/SWRL, last access on Dez 2008.

[12] L. Juszczyk, Hong-Linh Truong, and S. Dustdar. Genesis - a framework for automatic generation and steering of testbeds of complexweb services. pages 131 –140, mar. 2008.

[13] 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.

[14] Oliver Kopp, Branimir Wetzstein, Ralph Mietzner, Stefan Pottinger, Dimka Karastoyanova, and Frank Leymann. A Model-Driven Approach to Implementing Coordination Protocols in BPEL. In *1st International Workshop on Model-Driven Engineering for Business Process Management (MDE4BPM 2008)*, volume 17 of *Lecture Notes in Business Information Processing*, pages 188–199. Springer-Verlag, September 2008.

[15] Kyriakos Kritikos and Dimitris Plexousakis. Semantic QoS Metric Matching. In *ECOWS*, pages 265–274, 2006.

[16] Kyriakos Kritikos and Dimitris Plexousakis. Evaluation of QoS-Based Web Service Matchmaking Algorithms. In *International Conference on Services Computing*, 2008.

[17] Freddy Lécué and Alexandre Delteil. Making the difference in semantic web service composition. In *AAAI*, pages 1383–1388, 2007.

[18] Freddy Lécué and Alain Léger. A formal model for semantic web service composition. In *ISWC the 5th International Semantic Web Conference*, pages 385–398, November 2006.

[19] David Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Professional, May 2002.

[20] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic Markup for Web Services. Internet (http://www.daml.org/services/owl-s/1.0/owl-s.pdf), 2004.

[21] Sheila A. McIlraith and Tran Cao Son. Adapting golog for composition of semantic web services. In *KR*, pages 482–496, 2002.

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

[23] Elisabetta Di Nitto. State of the art report on software engineering design knowledge and survey of hci and contextual knowledge. Deliverable PO-JRA-1.1.1, S-Cube Consortium, July 2008. The following institutions contributed to this deliverable: Tilburg University, Center for Scientific and Technological Research, Politecnico di Milano, University of Duisburg-Essen and City University London.

[24] OASIS. Web Services Distributed Management: Management Using Web Services (WSDM-MUWS) 1.1. Standard Specification, 2006.

[25] Justin O'Sullivan, David Edmond, and Arthur H. M. ter Hofstede. What's in a service? *Distributed and Parallel Databases*, 12(2/3):117–133, 2002.

[26] M. Paolucci, T. Kawamura, T.R. Payne, and K. Sycara. Semantic matching of web services capabilities. In *ISWC*, pages 333–347, 2002.

[27] Peter F. Patel-Schneider. A Proposal for a SWRL Extension to First-Order Logic. Proposal, DARPA DAML Program, November 2004.

[28] Marco Pistore, Fabio Barbon, Piergiorgio Bertoli, D. Shaparau, and Paolo Traverso. Planning and monitoring web service composition. In *AIMSA*, pages 106–115, 2004.

[29] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1:77–106, 2005.

[30] Daniel Schall, Hong-Linh Truong, and Schahram Dustdar. Unifying Human and Software Services in Web-Scale Collaborations. *IEEE Internet Computing*, 12(3):62–68, 2008.

[31] Quan Z. Sheng, Boualem Benatallah, Zakaria Maamar, and Anne H. H. Ngu. Configurable composition and adaptive provisioning of web services. *IEEE T. Services Computing*, 2(1):34–49, 2009.

[32] Keith D. Swenson, Sameer Pradhan, and Mike D. Gilger. Wf-XML 2.0 XML Based Protocol for Run-Time Integration of Process Engines. Technical report, WfMC, 2004.

[33] Katia P. Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan. Automated discovery, interaction and composition of semantic web services. *J. Web Sem.*, 1(1):27–46, 2003.

[34] Martin Treiber. Overview of the state of the art in composition and coordination of services. Deliverable CD-JRA-2.2.2, S-Cube Consortium, March 2009. The following institutions contributed to this deliverable: University of Crete, Universidad Politécnica de Madrid, University of Stuttgart and Vienna University of Technology.

[35] Schall D. Plexousakis D. Treiber M., Kritikos K. and Dustdar S. Modeling context-aware and socially-enriched mashups. In *Third International Workshop on Web APIs and Services Mashups (Mashups'09) at OOPSLA 2009*, 2009.

[36] Tammo van Lessen, Frank Leymann, Ralph Mietzner, Jorg Nitzsche, and Daniel Schleicher. A management framework for ws-bpel. *Web Services, European Conference on*, 0:187–196, 2008.

[37] M. Vasko and S. Dustdar. Introducing Collaborative Service Mashup Design. In *Lightweight Integration on the Web (ComposableWeb'09)*, pages 51–62. CEUR - Workshop Proceedings, June 2009.

[38] Rainer von Ammon. Event-driven business process management. In *Encyclopedia of Database Systems*, pages 1068–1071. Springer, 2009.

[39] B. Wetzstein, P. Leitner, F. Rosenberg, I. Brandic, S. Dustdar, F. Leymann, and G. Stuttgart. Monitoring and Analyzing Influential Factors of Business Process Performance. In *Proceedings of the IEEE International Enterprise Distributed Object Computing Conference (EDOC 09)*, pages 141–150, 2009.

[40] Branimir Wetzstein. Overview of the state of the art in composition and coordination of services. Deliverable PO-JRA-2.2.1, S-Cube Consortium, July 2008. The following institutions contributed to this deliverable: University of Crete, Universidad Politécnica de Madrid, University of Stuttgart and Vienna University of Technology.

[41] Branimir Wetzstein, Dimka Karastoyanova, Oliver Kopp, Frank Leymann, and Daniel Zwink. Cross-Organizational Process Monitoring based on Service Choreographies. In *Proceedings of the 25th Annual ACM Symposium on Applied Computing (SAC 2010); Sierre, Switzerland, 21-26 March, 2010*. ACM, March 2010.

[42] Sonja Zaplata, Dirk Bade, Kristof Hamann, and Winfried Lamersdorf. Ad-hoc management capabilities for distributed business processes. In *submitted to: 3rd International Conference on Business Process and Services Computing (BPSC 2010)*, 9 2010.

A Attached Papers

A.1 The Frame Problem in Web Service Specifications

Authors:

UOC: George Baryannis

UOC: Dimitris Plexousakis

- Submitted to: PESOS 09 @ ICSE