Grant Agreement N° 215483

Title:	*QoS and SLA Aware Service Runtime Environment*
Authors:	*TUW, UOC, INRIA, SZTAKI*
Editors:	*Philipp Leitner , Harald Psaier (TUW)*
Reviewers:	*Pierluigi Plebani (POLIMI), Annapaola Marconi (FBK)*
Identifier:	*CD-JRA-2.3.9*
Type:	*Contractual Deliverable*
Version:	*1.0*
Date:	*28 Feb 2011*
Status:	*Final*
Class:	*External*

Management Summary

This deliverable contains the final research outcomes of work package WP-JRA-2.3 (Self-* Service Infrastructure and Service Discovery Support). Hence, most focus is set on research in the area of service registries, autonomic service runtime environments and non-functional aspects of service-based systems. The deliverable is a paper-based document, integrating results from 7 individual research papers, authored by various members of WP-JRA-2.3. This deliverable outlines the individual research, and puts the conducted work in the broader context of the S-Cube framework, outlining clearly how the individual partner research relates to other work conducted in the work package, as well as to results produced in different parts of the S-Cube project. Additionally, we also given an outlook on open issues and future research directives, which have been opened up by the work presented here.

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

Chapter 1

Deliverable Overview

1.1 Introduction

This deliverable presents the final S-CUBE research outcomes in work package WP-JRA-2.3 (Self-* Service Infrastructure and Service Discovery Support). More concretely, this document presents final results of task T-JRA-2.3.2 (Service Registration and Search), which deals mainly with service infrastructures and service discovery. The goals of the deliverable, as stated in the most recent version of the S-CUBE Description of Work, are as follows:

> *CD-JRA-2.3.9: QoS and SLA aware service runtime environment [Month 48]: The main goal of this work is to propose a description of a novel service runtime infrastructure, which will incorporate an active and QoS-aware registry and client components. This infrastructure will ensure SLA compliance and suggest services as well as ad hoc processes.*

Hence, this deliverable will be a research- and paper-oriented document, focusing mainly on the topics of Quality-of-Service (QoS) management and service discovery based on non-functional properties in the context of service registries and runtime environments. To this end, this deliverable builds on CD-JRA-2.3.3 [38], which presented some groundwork requirements and research challenges. The current deliverable is meant as a continuation and implementation of this more vision-oriented earlier document. Additionally, the current deliverable has to be seen as complementary to PO-JRA-2.3.7 [66], which considered ad hoc process detection based on events. Finally, as the scope of this deliverable is very closely related to non-functional properties, QoS and Service Level Agreement (SLA) management, this deliverable in particular has strong links to S-CUBE work package WP-JRA-1.3 (End-to-End Quality Provision and SLA Conformance). More concretely, the notion of end-to-end SLAs, as incorporated here, is discussed in more detail in CD-JRA-1.3.3 [33], CD-JRA-1.3.4 [55] and CD-JRA-1.3.5 [63].

1.2 Deliverable Structure

The remainder of this document is structured as follows. In Section 1.3, we will revisit the research architecture of WP-JRA-2.3. This section gives a coarse-grained overview over the general research work carried out in the work package in total, and concisely summarizes what parts of the work package vision have been covered in this document. In Section 1.4, we will introduce the problem area of this deliverable. Most importantly, we will revisit the notions of Non-Functional Properties (NFP), Quality-of-Service (QoS), QoS-based service discovery, and Service Level Agreements (SLAs). The following Section 1.5 gives a brief introduction to the collected contributions, their relation to other deliverables, and other Workpackages, and future directions opened by the work . Afterwards, Chapter 2 will contain individual discussions and overviews over the contributed papers. These sections give a quick glance over the relevant research and integration between research results. Section 3.1 will conclude the main

part of the deliverable with a short summary, and an outlook on future research directions and remaining open issues. Finally, Appendix A contains the original papers in verbatim. This allows the interested reader to dig into the contributed research in all details.

1.3 The WP-JRA-2.3 Research Architecture

Research work in WP-JRA-2.3 is driven by the Work Package vision that structures the research work internally. Figure 1.1 illustrates the overall research architecture of WP-JRA-2.3: research on service infrastructures is comprised in three threads, Service Discovery, Service Registries and Service Execution. Orthogonally different approaches are separated in three layers.

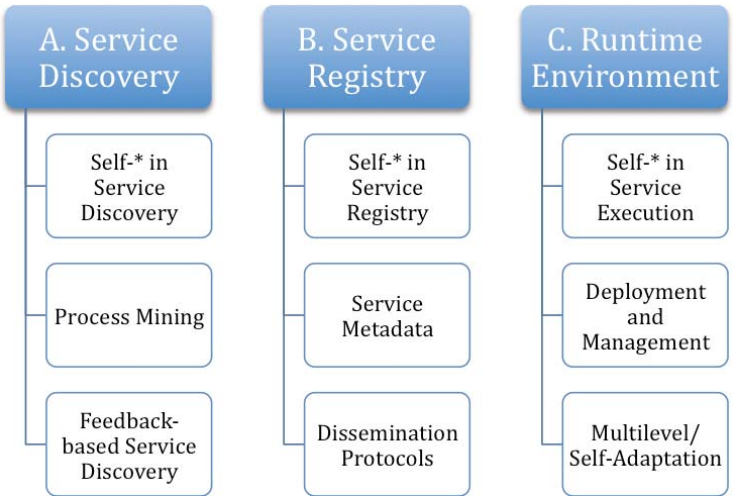

Figure 1.1: WP-JRA-2.3 Research Architecture

- Service Discovery Thread (A) - Service discovery is a fundamental element of service-oriented architectures, services heavily rely on it to enable the execution of service-based applications. Novel discovery mechanisms must be able to deal with millions of services. Additionally, these discovery mechanisms need to consider new constraints, which are not prevalent today, such as Quality of Experience requirements and expectations of users, geographical constraints, pricing and contractual issues, or invocability.
- Service Registry Research Thread (B) - Service registries are tools for the implementation of loosely-coupled service-based systems. The next generation of registries for Internet-scale service ecosystems are emerging, where fault tolerance and scalability of registries is of eminent importance. Autonomic registries need to be able to form loose federations, which are able to work in spite of heavy load or faults. Additionally, a richer set of metadata is needed in order to capture novel aspects such as self-adaptation, user feedback evaluation, or Internet-scale process discovery. Another research topic is the dissemination of metadata: the distributed and heterogeneous nature of these ecosystems asks for new dissemination methods between physically and logically disjoint registry entities, which work in spite of missing, untrusted, inconsistent and wrong metadata.
- Runtime Environment Research Thread (C) - There is an obvious need for automatic, autonomic approaches at run-time. As opposed to current approaches we envision an infrastructure that is able to adapt autonomously and dynamically to changing conditions. Such adaptation should be supported by past experience, should be able to take into consideration a complex set of conditions and their correlations, act proactively to avoid problems before they can occur and have a long lasting, stabilizing effect.

The current deliverable can be considered a cross-cutting, final discussion across all three research threads of the work package (service discovery, service registry and runtime environment). More concretely, the following aspects are covered by papers in this deliverable:

- We present some important contributions on research aspect A3 (feedback-based service discovery), related to skill-based discovery of human-provided services in marketplaces.
- In reserach thread B (service registry), we present important results on all three aspects (self-*, metadata and metadata dissemination). However, most focus is set on aspect B2 (service metadata).
- Finally, we also present results with regard to self-* in service execution (C1) and multi-level adaptation (C3), for instance within the scope of the PREvent framework.

However, please note that, as this is the final deliverable in this work package, the overal focus of the work presented was of an integrative nature, i.e., less focus was put on covering single aspects from the WP-JRA-2.3 research agenda, and more on cross-cutting research, spanning more than one aspect. Additionally, most research discussed in this deliverable also relates to other work packages of S-CUBE (most importantly, WP-JRA-1.2, WP-JRA-1.3 and WP-JRA-2.2).

1.4 Background

In the following section, we briefly summarize the most important background ideas used in the deliverable. This includes the idea of service metadata, non-functional service properties, Quality-of-Service, service discovery and Service Level Agreements (SLAs).

1.4.1 Non-Functional Properties and Quality-of-Service

Service descriptions include metadata about services. At least three different groups of metadata can be differentiated:

- *Functional descriptions* are the most common metadata, and define the functionality that a service provides. Simple functional descriptions can be published using WSDL [75]. More complex service descriptions are often specified using semantic Web service [47] technology, for instance WSMO [58] or SAWSDL [32]. However, there are also approaches which do not rely on semantics to provide more powerful functional descriptions [60].
- *Protocol descriptions* cover the dynamic aspects of service description. These are only relevant for stateful Web services, where service invocations have to be issued following a defined protocol or Message Exchange Patterns. These “usage protocols” for stateful services can be specified using languages such as BPEL Light [43] or the SEPL [20].
- Finally, QoS [48] aspects denote the non-functional properties of services. Hence, QoS is often seen as a discriminator between functionally equivalent services.

The abstract concept of QoS can be measured in virtually infinite different dimensions. Often-used dimensions to measure QoS are availability, response time, failure rate (reliability) and security. However, other research papers have identified a plethora of additional metrics. For instance, in [61] the dimensions wrapping time (time to unwrap the request XML structure), execution time (time necessary for the actual service invocation, excluding networking and message serialization issues) and network latency are proposed. Others consider trust and reputation as qualities that a service can exhibit [73, 68], and that can be used to differentiate between services.

1.4.2 Service Discovery Based on Non-Functional Properties

Service discovery is a process in which a service request is matched with the service descriptions/advertisements stored in a service registry. The result of this process is a set of service advertisements that match the request which may be grouped into a set of categories depending on their degree of match with this request. Based on the service aspect considered, this process can separated into two main sub-processes: a) functional service discovery, and b) non-functional service discovery. In the former process, the requester's functional requirements are matched with the functional capabilities of the registered services so as to infer those services that functionally match the request.

The non-functional service discovery process is usually called after the functional one. It is usually separated into two sequential sub-processes: a) non-functional service matchmaking, and b) service selection. The non-functional service matchmaking process filters the functional matching services based on their non-functional capabilities with respect to the non-functional requirements of the requester. It may also group the matching results based on their matching degree with the non-functional part of the service request. It must be noted here that both non-functional service capabilities and requirements are usually expressed as a set of constraints on non-functional properties and metrics. The service selection process then orders the matchmaking results according to their rank, which is usually produced through the Simple Additive Weighting [21] technique by considering requester-provided weights on non-functional terms (i.e., properties and metrics) as well as non-functional term-specific utility functions.

Research work in non-functional service matchmaking can be categorized into three main categories: ontology-based, constraint-based, and mixed. Ontology-based approaches [78] rely on semantic non-functional service specifications and use reasoners to infer the matching between non-functional service requests and advertisements. The main drawback of these approaches is that they are able to handle only n-ary constrained non-functional service specifications, i.e., specifications containing constraints on only one non-functional term. This drawback is solved by the constraint-based approaches [13, 11] which combine a non-functional service request and advertisement into one or more constraint models and use constraint solvers to solve these models and infer if there is a match between the non-functional request and advertisement. Depending on the linearity of the constraints involved in the specifications, different constraint solving techniques can be used [36]: a) Mixed-Integer Programming [65] for specifications containing only linear constraints and Constraint Programming [62] for specifications also containing non-linear constraints. However, such approaches assume that the non-functional service specifications reference specific non-functional models (i.e., descriptions of non-functional terms) that have been produced by one or more experts. The mixed-approach [36] relies on semantic non-functional service specifications and uses a specific algorithm [34] to align them based on their quality terms. It then follows the constraint-based approach to infer the matching of the specifications.

Apart from the previously described main approach in service selection, other sophisticated approaches [45] perform normalizations and grouping of non-functional metrics (in domains or functional groups). Normalizations are performed for three main reasons: a) to allow for a uniform measurement of service qualities independent of units, b) to provide a uniform index to represent service qualities for each provider, and c) to allow setting a threshold regarding the qualities. The number of performed normalizations depends on the nesting degree of the non-functional metric groups. All previous approaches rely on non-functional service advertisements that specify equality constraints on non-functional metrics (e.g. that the average response time is equal to 5 seconds). However, as in reality the non-functional service advertisements specify a range of values for each metric (i.e., two constraints defining the upper and lower value of a metric), approaches [11, 34] that solve Constraint Satisfaction Optimization Problems (SCOP) have been proposed to produce the rank for a non-functional service advertisement based on the worst (or even best) allowed value for the involved metrics.

1.4.3 Service Level Agreements

In a way, Web service SLAs [28] are a formalization and contractual arrangement of the concept of QoS. Instead of assuming that services provide the highest quality they can on a best-effort basis, SLAs fix the minimally promised quality in various dimensions. SLAs are often seen as legally binding contracts between one or more service clients and a service provider. SLAs are mainly a collection of SLOs. An SLO is an observable quality dimension of a service. Evidently, there is a strong relationship between SLOs and QoS, and, indeed, frequently SLOs are defined on top of QoS characteristics. For instance, often-used SLOs are the response time and availability of the service. Additionally, SLAs define penalties for non-achievement (violation) of SLOs. Penalties are often monetary consequences, which are expected to press service providers to achieve their target values. Both, penalties and target values, can be different for every SLA in which a an SLO is used. At runtime, concrete values for SLOs can be monitored. Based on the type of SLO (see below), this measured value can be generated either per composition instance or per aggregation interval.

Some different languages have been proposed to model SLA, including WSLA [12, 28], WS-Agreement [3] and SLAng [67]. These models do not differ so much in their expressiveness, but more in the environment they live in. For instance, WSLA specifies a monitoring and accounting infrastructure along with the basic language [12]. It is important to note that the work in this deliverable is agnostic with regard to the used SLA language, as long as it fits the basic model described above.

Types of SLOs

Just like QoS dimensions, SLOs come in different flavors. In this deliverable, two distinctions are of relevance.

- Firstly, one can differentiate between nominal and continuous SLOs. For nominal SLOs, the measured value of an objective can be one of a finite number of potential values. For these SLOs, the target value is a subset of the set of potential values. Metric SLOs, which are more prevalent, can take an infinite number of values. Target values are defined as thresholds on the metric.
- Secondly, one can distinguish SLOs on composition instance level and aggregated SLOs. For composition instance level SLOs, a decision of whether an SLA violation has happened can be made for every single composition instance individually. Aggregated SLOs are defined over an aggregation period, for instance a number of composition instances or a time interval. Decisions can be made only looking at the whole aggregation period, i.e., usually numerous composition instances. Unless stated otherwise, all work in this deliverable is applicable for composition instance level SLOs only. A generalization of the presented approach to aggregated SLOs is part of ongoing work.

1.5 Overview of the Contributions

In the following, we briefly introduce the contributions of the deliverable. More detailled summaries can be found in Chapter 2.

1.5.1 Stimulating Skill Evolution in Market-based Crowdsourcing [64]

Content Overview. This work presented at the 9th International Conference on Business Process Management (2011) presents Crowdsourcing on top of a SOA related infrastructure A major challenge in crowdsourcing is to guarantee high-quality processing of tasks. The work presents a novel approach that matches tasks to suitable workers based on auctions. This way QoS constraints can be better met and agreements settled with customers.

Relations to 2.3. The work is related to platform, thus, infrastructure management (JRA-2.3). In particular, the content can be seen as an extension to the ranking approaches in CD-JRA-2.3.5. Instead of discussing ideas of local approaches such as in CD-JRA-2.3.4 and CD-JRA-2.3.6 here a more global approach is taken when adapting the environment's resource selection. The adaptation strategies presented do not consider direct interference with the infrastructure but instead choose an offline approach that feeds its decision information from interaction observations, e.g., skill evolution.

Relations to other workpackages. While the current version of the paper has little relations to other work packages, future workcould go in the direction of compositions as studied by JRA-2.2. In particular, coordinated service compositions as researched in CD-JRA-2.2.2 and algorithms and techniques for splitting and merging compositions as presented in CD-JRA-2.2.3 would then be in the focus of the studies.

Future Directions. As part of the ongoing research the plan is to investigate the difficulties to introduce complex tasks and in the crowd collaboration. Sub-tasks need to be decompose and reassembled. The responsibility for a certain QoS would also transfer partially to the crowd members.

1.5.2 End-to-End Support for QoS-Aware Service Selection, Binding and Mediation in VRESCo [50]

Content Overview. Published in IEEE Transactions on Services Computing, this work copes with the challenge of dynamic and adaptable service-oriented solutions with special emphasis on service metadata, Quality of Service, service querying, dynamic binding and service mediation. The Vienna Runtime Environment for Service-oriented Computing (VRESCo) is presented in the light of a service infrastructure with details on service querying and service mediation.

Relations to 2.3. This work is integral to to research agenda of JRA-2.3 (service registries), as it presents a novel service registry model, which is particularly suitable to support adaptive SOAs. Service querying in VRESCo builds on the ranking approaches discussed in CD-JRA-2.3.5.

Relations to other workpackages. As VRESCo is, by design, QoS-aware, this contribution also strongly relates to JRA-1.3, most importantly to the QoS evaluation methods discussed in CD-JRA-1.3.4, CD-JRA-1.3.5 and CD-JRA-1.3.6.

Future Directions. Future work within VRESCo will see us focus more on support and integration of service compositions and business processes. Hence, this line of research will mostly continue within the topics covered by JRA-2.2.

1.5.3 Cost-Based Optimization of Service Compositions [40]

Content Overview. This work, accepted for publication in IEEE Transactions on Services Computing, tackles the challenge of preventing cases of SLA violations for providers of composite services. In order to get a realistic and complete view of the decision process of service providers, the costs of adaptation need to be taken into account. The solution presented offers possible algorithms to solve this complex optimization problem, and details an end-to-end system based on the PREvent (prediction and prevention based on event monitoring) framework, which clearly indicates the usefulness of the model.

Relations to 2.3. At its core, this contribution is discussing an autonomic service-based system for managing customer SLAs. To this end, we use the VRESCo QoS-aware service registry (see previous contribution) as foundation. Hence, this contribution relates to the self-* service registry and infrastucture part of JRA-2.3.

Relations to other workpackages. Evidently, this paper also has strong relationships to JRA-1.2, especially CD-JRA-1.2.4, CD-JRA-1.2.5 and CD-JRA-1.2.6. Furthermore, the contribution relates strongly to the SLA topics discussed in JRA-1.3 (CD-JRA-1.3.6). Finally, as the main subject of adaptation in this paper are service compositions, the paper also relates to JRA-2.2 (CD-JRA-2.2.6).

Future Directions. There is still potential for plenty of further research in the direction of cost-based optimization. For instance, in its current form, the formalization used in the paper does not take into account indirect costs, such as customer satisfaction or potential loss of future customers. Furthermore, the current version of the paper suffers from the limitation that it considers only SLAs on instance-level. Future work can extend and improve on those aspects.

1.5.4 Towards Optimizing the Non-Functional Service Matchmaking Time

Content Overview. Published as a poster contribution at the The World Wide Web Conference, the approach concentrates on exploiting constraint solving techniques in service discovery for inferring if the user non-functional requirements are satisfied by the service non-functional capabilities. This paper proposes two alternative techniques for improving the non-functional service matchmaking time. The first one is generic as it can handle non-functional service specifications containing n-ary constraints, while the second is only applicable to unary-constrained specifications.

Relations to 2.3. The research work addresses the JRA-2.3's research challenge that concerns scalable and fault tolerant techniques for service discovery as it proposes two QoS-based service matchmaking techniques which not only optimize the matchmaking time without sacrificing accuracy but they can also be distributed through assigning part of the matchmaking functionality to different nodes.

Relations to other workpackages. The research work is highly connected to the JRA-1.3 WP as it exploits non-functional service descriptions (requests as well as advertisements), which could be described through the non-functional service meta-model proposed in the deliverable CD-JRA-1.3.3 and could reference the non-functional properties of the S-Cube's end-to-end quality reference model proposed in the deliverable CD-JRA-1.3.2, so as to perform the matchmaking.

Future Directions. The research work can be exploited by scalable service registries which are able to matchmake millions of services, paving the way for the move towards the Internet of Services.

1.5.5 Cost Reduction Through SLA-driven Self-Management [37]

Content Overview. Presented at the 9th IEEE European Conference on Web Services, the presented approach considers a SLAs management which satisfies the customers requirements and also their own business objectives, such as maximizing profits. Most current systems fail to consider business objectives and thus to provide a complete SLA management solution. Specifically, this work proposes a framework that comprises multiple, configurable control loops and supports automatically adjusting service configurations and resource usage in order to maintain SLAs in the most cost-effective way. The framework targets services implemented on top of large-scale distributed infrastructures, such as clouds.

Relations to 2.3. The work is placed in the context of *Self-* in Service Execution*. Qu4DS provides an automatic support for executing services by using self-adaptive techniques (dynamic adaptation). With regard to *Deployment and Management*, the Qu4DS framework automatically deploys service instances on top of distributed infrastructures by managing them according to quality properties.

Relations to other workpackages. The work is complementary to adaptable service compositions (i.e., WP-JRA-2.2) as it prevents SLA violations thus avoiding triggering adaptation actions in the

composition-level. Moreover, composition-level adaptable techniques can be aware about further details about Qu4DS adaptive behavior which enables to conceive multilevel adaptation. With respect to the deliverable CD-JRA-2.3.2, the work can be considered as a self-healing support for atomic services. With respect to the deliverable CD-JRA-2.3.6, Qu4DS specifies and employs adaptation policies but not limit to them, the framework can be extended to support further adaptation policies.

Future Directions. The work leaves place for several research directions. First, the QoS translation can be improved based on advanced application profiling techniques. This may require estimating application performance based on profiled data at runtime for instance. Second, Qu4DS could rely on dynamic pricing where the price of the service take into account further aspects, for example, related to competitor service prices. Third, other service providers can be supported, not only those which rely on distributed tasks. This requires further dynamic metrics which are able to analyze if the request treatment is in time or delayed for example. Finally, further adaptation policies can be developed not only in the sense of the current addressed events (request arrivals and job faults and delays), but also in the context of other events such as contract proposals, resource failures, infrastructure price changes, contract rescission and so forth.

1.5.6 Autonomic SLA-aware Service Virtualization for Distributed Systems [30]

Content Overview. Presented at the 19th Euromicro International Conference on Parallel, Distributed and Network-Based Computing, the focus in this work is on Cloud Computing. Managing such heterogeneous environments requires sophisticated interoperation of adaptive coordinating components. The work introduces an SLA-aware Service Virtualization architecture that provides non-functional guarantees in the form of SLAs and consists of a three-layered infrastructure including agreement negotiation, service brokering and on demand deployment. In order to avoid costly SLA violations, flexible and adaptive SLA attainment strategies are used with a failure propagation approach.

Relations to 2.3. The work introduced in this paper is a result of a SZTAKI-TUW collaboration in the Runtime Environment research thread of JRA-2.3. It targets the management and deployment aspects of interoperating distributed systems, and proposes autonomic SLA management strategies to deal with the highly dynamic and error-prone nature of these systems. In the S-Cube deliverable CD-JRA-2.3.2, a conceptual architecture incorporating three closely related areas was introduced: agreement negotiation, service brokering and service deployment. The examinations of this architecture revealed the basic requirements for a future self-* realization of these core components. These basic requirements implied that there must be a negotiation phase when it is specified, what service is to be invoked and what are the conditions and constraints (temporal availability, reliability, performance, cost, etc.) of its use. The contribution to Deliverable CD-JRA-2.3.4 refines this vision, and presents a unified service virtualization environment representing the first attempt to combine SLA-based resource negotiations with virtualized resources in terms of on-demand service provision resulting in a holistic virtualization approach. The contributino to Deliverable CD-JRA-2.3.6 introduces the autonomic behaviour of this service virtualization architecture by summarizing the same contribution. The focus is on the SLA-awareness of the architecture.

Relations to other workpackages. The topic of this contribution is also closely related to research carried out in WP-JRA-1.2 and WP-JRA-1.3. Detailed adaptation-related capabilities of the architecture including cross-layer adaptation were reported in deliverable CD-JRA-1.2.5.

Future Directions. In this current contribution a refined architecture was introduced considering resource provision using a virtualization approach combined with the business-oriented utilization. This solution utilizes a heterogeneous SLA-coupled infrastructure for on-demand service provision based on SLAs with a MAPE-based autonomic behaviour, in order to cope with changing user requirements and on demand failure handling. Since newly emerged technologies such as Cloud Computing become

more and more utilized in business service solutions, the extension of the proposed SSV solution for additional Cloud infrastructures and platforms will require further research.

The volume of works collected in this deliverable present different approaches to the subject of the deliverable. The particular challenges studied in the partners' contributions describe approaches taken from different perspectives and taking into account various relevant aspects of the Service-Oriented Architecture (SOA). In the next section, all partners present their contribution in the scope of this deliverables description.

Chapter 2

Contributions to QoS and SLA aware service runtime environment

2.1 Stimulating Skill Evolution in Market-based Crowdsourcing

Contributing partners: Vienna University of Technology (TUW)

Status: Submitted to the 9th International Conference on Business Process Management (BPM) , 28th August - 2nd September, 2011, Clermont-Ferrand, France.

Keywords: Human-centric BPM, Crowdsourcing, Online communities, Task markets, Auctions, Skill evolution

2.1.1 Background

Today, ever changing requirements force in-house business processes to rapidly adapt to changing situations in order to stay competitive. Changes involve not only the need for process adaptation, but also, require an additional inclusion of new capabilities and knowledge, previously unavailable to the company. Thus, outsourcing of parts of business processes became an attractive model. This work, in particular, focuses on a distinguished recent type of outsourcing called *crowdsourcing*. The term crowdsourcing describes a new web-based business model that harnesses the creative solutions of a distributed network of individuals [8], [72]. This network of humans is typically an open Internet-based platform that follows the *open world* assumption and tries to attract members with different knowledge and interests. Large IT companies such as Amazon, Google, or Yahoo! have recognized the opportunities behind such *mass collaboration systems* [14] for both improving their own services and as business case. In particular, Amazon focuses on a task-based marketplace that requires explicit collaboration. The most prominent platform they currently offer is *Amazon Mechanical Turk* (AMT) [2]. Requesters are invited to issue *human-intelligence tasks* (HITs) requiring a certain qualification to the AMT. The registered customers post mostly tasks with minor effort that, however, require human capabilities (e.g., transcription, classification, or categorization tasks [22]).

2.1.2 Problem Statement

In this work we assume that crowdsourcing can be build on top of a SOA environment. The main challenges addressed in this work relate to building and managing an automated crowd platform. There is only a few approaches towards integrating SLAs into crowdsourcing. In one of our previous works at TUW [57] we introduced an approach to include human tasks as available on the current crowd platforms into Web Service Level Agreements (WSLA). The interested reader is referred to the results of that

work for a more elaborated study of the idea. This present work relates more to QoS related issues. In crowdsourcing it is not only of importance to find suitable workers for a task and to provide the customer with satisfying quality, but also, to maintain a motivated base of crowd members and provide stimulus for learning required skills. Only a recurring, satisfied crowd staff is able to ensure high QoS and high output. As any crowd, fluctuations must be compensated and a *skill evolution* model must support new and existing crowd workers in developing their capabilities and knowledge. Finally, the standard processes on such a platform should be automated and free from intervention to handle the vast amount of tasks and to make it compatible with a SOA approach. Atop, the model should increase the benefit of all participants to support QoS.

2.1.3 Contribution Relevance

As outlined previously, a further major challenge hampering the establishment of a new service-oriented computing paradigm spanning enterprise and open crowdsourcing environments are quality issues. In our scenario this is strongly connected to correctly estimating the skills of workers. Thus, the presented **skill evolution** approach helps to increase the confidence in worker skills with qualification tasks. This would imply a huge overhead for the testing requester; s/he is also the only one who benefits from the gathered insights. Here, we take a different approach by integrating the capability of confidence management into the crowdsourcing platform itself. Instead of having point-to-point tests, we propose the automated assessment of workers to unburden requesters in inspecting workers' skills. The approach is *(semi-)automatic* and offers great potential for inclusion of crowd capabilities in business environments. Knowing crowd capabilities by *skill evolution* is one of the major steps towards better QoS in an open crowdsourcing environment.

2.1.4 Contribution Summary

Our idea of skill evolution contains the following steps. We propose the *automatic assessment* of workers where confidence values are low. For example, newcomers who recently signed up to the platform may be high or low performers. To unveil the tendency of a worker, we create a hidden 'tandem' task assignment comprising a worker whose skills are known (high performer) with a high confidence and a worker where the crowdsourcing platform has limited knowledge about its skills (i.e., low confidence). The next step is that both workers process the *same* task in the context of a requester's (real) task. However, only the result of the high confidence worker is returned to the requester, whereas the result of the low confidence worker is compared against the delivered reference.

This approach has advantages and drawbacks. First, skill evolution through tandem assignments provides an elegant solution to avoid training tasks (assessments are created automatically and managed by the platform) and also implicitly stimulates a learning effect. Of course, the crowdsourcing platform cannot charge the requester for tandem task assignments since it mainly helps the platform to better understand the true skill (confidence) of a worker. Thus, the platform must pay for worker assessments. As the evaluations show, performing assessments provides the positive effect that *the overall quality* of provided results and thus requester *satisfaction increases* due to a better understanding of worker skills.

2.1.5 Contribution Evaluation

To evaluate the ideas of the work we have implemented a Java-based simulation framework that supports all introduced concepts. An evaluation scenario consists of a set of workers and a set of requesters. In every round of the simulation each requester usually announces a task. An auction is conducted for each announced task which contains among other properties the expected quality. High quality requirements indicate highly sophisticated and demanding tasks. We applied different scenarios and tested those with and without skill evolution. Detailed description of the setup is available in the work's section on implementation an evaluation.

The results show that the additional assessments provide remarkable good results for reducing misjudgements. As a summary, skill evolution generally performs better, however, is not antagonistic to overload scenarios. While with moderate task offering frequencies the model performs much better in all measurements, the differences become even when load increases and assessment task further overload the platform. The results show that it is the responsibility of the platform to balance the task load and trade only with a fair amount of requesters.

2.1.6 Conclusions

In the contribution we present a novel crowdsourcing marketplace based on auctions. The design emphasizes automation and low overhead for users and members of the crowdsourcing system. In oder to increase the QoS in SOA-based crowdsourcing we introduce an approach with two novel auctioning variants and show by experiments that it may be beneficial to employ assessment task in order to estimate members' capabilities and to train skills. Stimulating the demand for certain skills in such a way leads to skill evolution. From such an evolution can benefit all participants. New workers can be motivated by helping them to develop their skills with training. This way the platform provider can harvest a regular group of returning workers. The customers are also satisfied because the quality of the service because the auctioning mechanism guarantees that the best fitting worker gets their tasks.

2.2 End-to-End Support for QoS-Aware Service Selection, Binding and Mediation in VRESCo

Contributing partners: Vienna University of Technology (TUW)

Status: Published in IEEE Transactions on Services Computing (2010)

Keywords: Web Services Publishing and Discovery, Metadata of Services Interfaces, Advanced Services Invocation Framework

2.2.1 Background

During the last years, Service-oriented Architecture (SOA) and Service-oriented Computing (SOC) have gained acceptance as a paradigm for addressing the complexity that distributed computing generally involves. In theory, the basic SOA model consists of three actors that communicate in a loosely coupled way as shown in Figure 2.1a. However, practice has shown that SOA solutions are often not as flexible and adaptable as claimed. We argue that there are some issues in current implementations of the SOA model. First and foremost, service registries such as UDDI did not succeed. We think this is partly due to their limited querying support that only provides keyword-based matching of registry content, and insufficient support for metadata and non-functional properties of services. This is also highlighted by the fact that Microsoft, SAP, and IBM have finally shut down their public UDDI registries in 2005. As a result, service registries are often missing in service-centric systems (i.e., no *publish* and *find* primitives). This leads to point-to-point solutions where service endpoints are exchanged at design-time (e.g., using E-mail) and service consumers statically bind to them (see Figure 2.1b). Besides that, support for dynamic binding and invocation of services is often restricted to services having the same technical interface. In this regard, the lack of service metadata makes it difficult for service consumers to know if two services actually perform the same task. Furthermore, support for Quality of Service (QoS) is necessary to enable service selection based on non-functional QoS attributes such as response time (in addition to functional attributes).

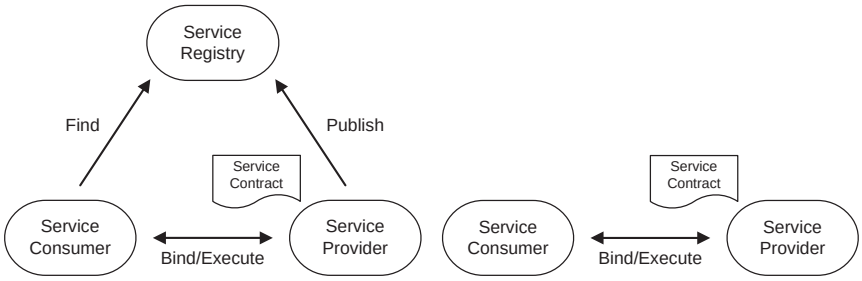

Figure 2.1: SOA Theory vs. Practice (adapted from [51])

2.2.2 Problem Statement

Adaptive service-oriented systems bring along several distinct requirements, leading to a number of challenges that have to be addressed. In this section, we summarize the current challenges we see most important, and which are addressed in the paper summarized here. (1) Firstly, service interface description languages such as WSDL focus on the interface needed to invoke a service. However, from this interface it is often not clear what a service actually does, and if it performs the same task as another service. Service metadata [7] can give additional information about the purpose of a service and its interface (e.g., pre- and post-conditions). (2) Once services and associated metadata are defined, this information should be discovered and queried by service consumers. This is the focus of service registry standards such as UDDI. In practice, the service registry is often missing. (3) In enterprise scenarios, QoS plays a crucial role [77] in discriminating between services. This includes both network-level attributes (e.g., latency and availability), and application-level attributes (e.g., response time and throughput). (4) Finally, in order to technically enable actual dynamicity and runtime service binding, mechanisms that mediate between alternative services, possibly having different interfaces, need to be provided.

2.2.3 Contribution Relevance

In order to tackle the four challenges outlined above, we have devised and implemented a service runtime environment called VRESCo, which aimed at providing a practical, integrative solution to adaptive enterprise services computing. To be more specific, the present paper focuses on service metadata, QoS and service querying, plus dynamic binding, invocation, and mediation of services. Additionally, we provide an extensive performance evaluation of the different components and an end-to-end evaluation of the overall runtime, that shows the applicability of our approach.

2.2.4 Contribution Summary

The architectural overview of VRESCo is shown in Figure 2.2, which is adapted from [49]. The VRESCo core services are provided as Web services that can be accessed either directly using SOAP or by using the Client Library that provides a simple API. Furthermore, the DAIOS framework [41] has been integrated into the Client Library, and provides stubless, protocol-independent, and message-driven invocation of services. The Access Control Layer guarantees that only authorized clients can access the core services, which is handled using claim-based access control and certificates [49]. Services and associated metadata are stored in the Registry Database which is accessed using the Object-Relational Mapping (ORM) Layer. Finally, the QoS Monitor is responsible for regularly measuring the current QoS values. The overall system is implemented in C# using the Windows Communication Foundation [46]. Due to the platform-independent architecture, the Client Library can be provided for different platforms (e.g., C# and Java).

There are several core services. The Publishing/Metadata Service is used to publish services and metadata into the Registry Database. Furthermore, the Management Service is responsible for managing

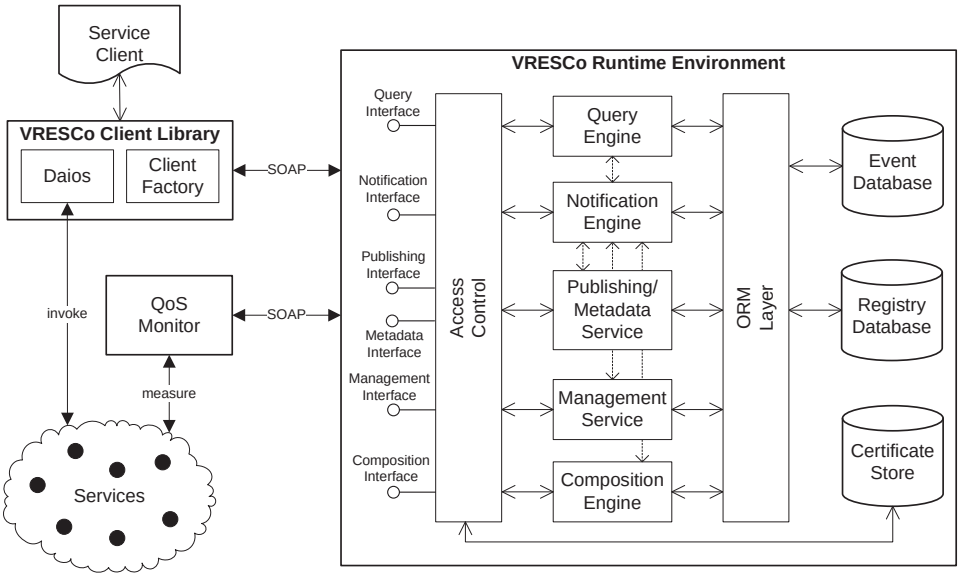

Figure 2.2: VRESCo Overview Architecture

user information (e.g., name, password, etc.) whereas the Query Engine is used to query the Registry Database. The Notification Engine informs users when certain events of interest occur inside the runtime, while the Composition Engine [59] provides mechanisms to compose services by specifying hard and soft constraints on QoS attributes. In this paper, we focus on the main requirements for our client-side mediation approach which are the Metadata Service (including the models for metadata, services and QoS), the Query Engine, and the dynamic binding, invocation and mediation mechanisms.

2.2.5 Contribution Evaluation

We give an evaluation of the VRESCo runtime focusing on the topics covered in this paper. The purpose of this evaluation is twofold: firstly, we show the runtime performance regarding service querying, rebinding, and mediation by using synthetic data. The main goal of this evaluation is to analyze the performance impact of each aspect in isolation. Secondly, we combine these aspects into a coherent end-to-end evaluation using an order processing workflow. The main goal is to understand the influence of each aspect with regard to the overall process duration in a realistic setting. Additionally, we show how the individual results of the first part interrelate in an end-to-end setting. For mediation, rebinding and end-to-end evaluation we have created different sets of test services and QoS configurations (with varying response times) using the Web service generation tool GENESIS [26].

2.2.6 Conclusions

One of the main promises of SOC is the provisioning of loosely-coupled applications based on the publish-find-bind-execute cycle. In practice, however, these promises can often not be kept due to the lack of expressive service metadata and type-safe querying facilities, explicit support for QoS, as well as support for dynamic binding and mediation. In this paper, we have proposed the QoS-aware VRESCo runtime environment which has been designed with these requirements in mind.

2.3 Cost-Based Optimization of Service Compositions

Contributing partners: Vienna University of Technology (TUW)

Status: Accepted for publication in IEEE Transactions on Services Computing (2012)

Keywords: Service Composition, Service Level Agreements, Adaptation, Optimization

2.3.1 Background

Service-based applications have seen tremendous research activity in the last years, with many important results being generated around the world. However, to fully realize its potential, research and industry alike need to focus more strongly on non-functional properties and quality issue of services. In the business world, QoS promises are typically defined within legally binding Service Level Agreements (SLAs) between clients and service providers, represented, e.g., using WSLA. SLAs contain Service Level Objectives (SLOs), i.e., concrete numerical QoS objectives, which the service needs to fulfill. If SLOs are violated, agreed upon monetary consequences go into effect. For this reason, providers generally have a strong interest in monitoring SLAs and preventing violations, either by using post mortem analysis and optimization [74], or by runtime prediction of performance problems [39]. We argue that the latter is more powerful, allowing to prevent violations before they have happened by timely application of runtime adaptation actions.

2.3.2 Problem Statement

However, preventing SLA violations is, in general, not for free. For instance, some alternative services usable in a composition may provide faster response times (thereby improving the end-to-end runtime of the composite service, and reducing the probability of violating runtime related SLOs), but those services are often more expensive than slower ones. Therefore, there is an apparent tradeoff between preventing SLA violations and the inherent costs of doing so. We argue that this tradeoff is currently not covered sufficiently in research. Instead, researchers assume that the ultimate goal of service providers is to minimize SLA violations, completely ignoring the often significant costs of doing so (e.g., [42, 27]).

2.3.3 Contribution Relevance

In this paper, we contribute to the state of the art by formalizing this tradeoff as an optimization problem, with the goal of minimizing the total costs (of violations and applied adaptations) for the service provider. We argue that this formulation better captures the real goals of service providers. Additionally, we present possible algorithms to solve this optimization problem efficiently enough to be applied at composition runtime.

2.3.4 Contribution Summary

In the paper, we model the decision process of a service provider as in Equation 2.1. Therein, TC are the total costs for the service provider; S contains all SLOs contained in the SLA of the provider; e^i_{sx} is an estimation function capturing the predicted costs for SLA violations; A^* is the set of applied adaptation actions (from the set A of potential adaptations); $c(a_x)$ are the costs of applying adaptation action a_x; finally, $v(A^*)$ is a penalty term that captures whether A^* contains any conflicting adaptation actions. For more details, please refer to the original paper.

$$TC(A^*) \approx v(A^*) + \sum_{s_x \in S} e^i_{sx} + \sum_{a_x \in A^*} c(a_x) \rightarrow min! \tag{2.1}$$

In addition to the optimization problem formalization, we also discuss different approaches for finding solutions to this problem. Firstly, we present a deterministic branch-and-bound approach, which is guaranteed to find the optimal solutions. However, given the problem structure and the generally very large solution space, for many problem instances a deterministic solution is unfeasible. For these cases,

we present two alternative heuristic approaches: a local search algorithm, which finds solutions very efficiently, and a genetic algorithm based approach, which takes more time to find solutions but is able to evade local optima.

2.3.5 Contribution Evaluation

In the paper, we evaluated our approach based on an illustrative example, which has been implemented using .NET Windows Communication Foundation[1] (WCF) technology and the VRESCo SOA runtime environment on a server running Windows Server Enterprise 2007, Service Pack 2. The machine was equipped with 2 2.99GHz Xeon X5450 processors and 32 GByte RAM. More details on the experimental setup can be found in the accompanying experimentation web page[2].

Drawing conclusions from our experiments, we note that Branch-and-Bound is applicable in situations where just a small set of actions is available. If more actions are available, Memetic Algorithms and GRASP are interesting candidate algorithms. GRASP produces good solutions in very little time and can generally be used even for short-running compositions where adaptation decisions need to be taken in a short time frame (below 1 second). Memetic Algorithms are very promising in case of long-running compositions, where the time necessary to find a solution is not critical. Memetic Algorithms often produce slightly better solutions than GRASP, but take much more time to do so. In a second set of experiments, we also evaluated the end-to-end effectiveness of our system. That is, we analyze if the system fullfills its main promise, preventing SLA violations and reducing the total costs for the service provider. As can be seen in the paper, our system fulfills its main promise: in the example case, the total number of SLO violations decreases to about 28% of the number of predicted violations. However, we can also see that it does not primarily prevent violations, but rather aims at minimizing the costs of violations. Thereby, the total costs for the service provider can be reduced to 56% of the predicted costs.

2.3.6 Conclusions

For providers of composite Web services, it is essential to be able to minimize cases of SLA violations. One possible route to achieve this is to predict at runtime, which instances are in danger of violating SLAs, and to apply various adaptation actions to these instances only. However, it is not trivial to identify which adaptations are the most cost-effective way to prevent any violation, or if it is at all possible to prevent a violation in a cost-effective way. In this paper, we have modelled this problem as a one-dimensional, discrete optimization problem. Furthermore, we have presented both, deterministic and heuristic solution algorithms. We have evaluated these algorithms based on a manufacturing case study, and shown which types of algorithms are better suited for which scenarios.

2.4 Towards Optimizing the Non-Functional Service Matchmaking Time

Contributing partners: University of Crete

Status: conditionally accepted (as a poster paper) at WWW'12

2.4.1 Background

One of the main drivers and mechanisms towards the Internet of Services (IoS), where millions of services will be available to users through a converged information, communication, and service infrastructure, is service-orientation as it promises the automatic construction of novel, added-value applications through the discovery and composition of services. However, service-orientation has not yet kept up

[1] http://msdn.microsoft.com/en-us/library/ms735967(VS.90).aspx
[2] http://www.infosys.tuwien.ac.at/prototype/VRESCo/experimentation.html

to its promises as it relies on a specific architecture, in which the service broker role, associated to the discovery of services, has not been successfully fulfilled by the current implementations. Such implementations lack the appropriate scalability that is required for matching millions of services and rely on service discovery mechanisms that are not very efficient in terms of matchmaking time and accuracy.

Concerning the functional service discovery, the state-of-the-art research work [10, 56, 31] relies on using semantic I/O-based service descriptions and exploits Semantic Web techniques to perform the functional service matchmaking. It has been shown that such work has a very good matchmaking time but not a perfect accuracy as it does not rely on the service goals, expressed by service pre- and post-conditions, as such descriptions are not yet provided by the service providers. Another line of work [16, 69, 52, 35] has focused on scalability issues and on further optimizing the matchmaking time by appropriately organizing the service advertisement and query space.

2.4.2 Problem Statement

While the research status concerning functional service discovery is satisfactory, the same cannot be stated for non-functional service discovery. The state-of-the-art approaches [11, 36] in the latter sub-area exhibit perfect accuracy and have mainly focused on optimizing the time required to matchmake a single non-functional request-to-advertisement pair by exploiting appropriate constraint solving techniques. However, they have not yet focused on optimizing the overall non-functional matchmaking time. To this end, they can take significant time to match a set of hundreds or thousands of non-functional service advertisements against a non-functional service request, so they are not yet appropriate for the move to the IoS era. In addition to the above problem, there have not been approaches focusing on scalability issues.

Thus, there is a need for scalable non-functional service discovery techniques that can appropriately manage a vast amount of non-functional service advertisements and optimize the time needed to match them against non-functional service requests. If such techniques were coupled with those proposed in functional service discovery, then a better service broker implementation would have been realized, which could enable users and automated agents to discover those services that perfectly match their tasks both functionally and non-functionally in a timely manner and with the appropriate, if not perfect, accuracy.

2.4.3 Contribution Relevance

By exploiting the matchmaking metric of the approach in [36], this work partially closes the above gap by proposing two different matchmaking techniques which are able to optimize the overall non-functional service matchmaking time without sacrificing matchmaking accuracy. This is shown both theoretically and empirically by comparing these two novel techniques against the most prominent state-of-the-art one proposed in [36] in terms of matchmaking time. In fact, one of these two techniques, called ”Unary matching” technique, is shown to be far better than the other and the prominent one not only concerning matchmaking but also insertion, deletion, and update time. Another significant advantage of the proposed work is that both techniques can be easily distributed in order to realize scalable matchmaking mechanisms.

2.4.4 Contribution Summary

The first proposed technique, called “Subsumes matching” technique, relies on the “subsumes” type of matchmaking metrics and on the fact that if a non-functional service specification A subsumes another specification B then it will also subsume all the specifications that are subsumed by B. To this end, it organizes the non-functional service advertisement space in such a way that the number of non-functional request-to-advertisements pairs examined is less than that of the state-of-the-art approach. In particular, it constructs a forest of “subsumes” trees, where each node corresponds to a non-functional service

advertisement and a parent node in each tree subsumes all of its children nodes. In this way, when a service request is issued, it is compared against the nodes of each tree from the root until the leaves. However, if it is found that it subsumes a specific node, then there is no need to go further down to the node's children/descendants as the request will certainly match/subsume them.

It is obvious that this technique is quicker than the state-of-the-art one, as each one uses the same matchmaking metric but the first technique performs less comparisons. However, the construction and update of a forest of "subsumes" trees is more costly than the construction and the update of a list of service advertisements (as it is the case for the prominent approach). To this end, this technique exhibits a higher insertion, deletion, and update time with respect to the prominent approach. This technique can be distributed by assigning the responsibility of matching a subset of the subsumes trees to different nodes.

The second proposed technique (i.e., the "Unary matching" one) relies on similar techniques performed in functional service matchmaking [10] in order to appropriately organize the non-functional service advertisement space. In particular, it maintains for each non-functional metric/property an ordered set of limits, where each limit may correspond to one or more non-functional specifications containing a respective metric bound or equality constraint on the limit's value. To this end, when a non-functional service request is issued, each of its unary constraints are examined based on their containing metric. Depending on the metric bound and constraint type, a sub-part of the metric's ordered list of limits is examined so as to produce a list of the matching non-functional advertisements' URIs. For instance, if the request constraint is of the form: $X \leq a$, then the limits that are equal or less to a are examined and the URIs of the non-functional specifications that have constraints of the form $X \leq a_1$, or $X == a_1$, where $a_1 \leq a$, are collected. For each request constraint, its constructed URI list is concatenated with that of the previous constraint. If the URI list concatenation is empty, then the non-functional request does not have a matching advertisement. Otherwise, after the processing of the last request constraint, the final, concatenated URI list contains all the URIs of the advertisements that match the request.

The second technique is quicker than the other proposed technique as well as the prominent matchmaking approach not only in terms of matchmaking but also insertion, deletion, and update time. Moreover, due to the way it organizes the advertisement space, it can be easily distributed by assigning the responsibility of matching a sub-set of all non-functional metrics to different nodes. Its sole drawback is that it relies on exploiting only unary-constrained non-functional service specifications.

2.4.5 Contribution Evaluation

The two proposed techniques were both theoretically and empirically evaluated against the most prominent non-functional service matchmaking approach [36]. The results reported validate the comparison statements referenced in the previous subsection. In particular, the results showed that the "Unary matching" technique is far better than the other two techniques and more scalable. In addition, they showed that both proposed techniques significantly outperform the state-of-the-art technique in terms of matchmaking time.

2.4.6 Conclusions

This work has proposed two novel techniques that are able to optimize the overall non-functional service matchmaking time with respect to the state-of-the-art research work. The "Unary matching" proposed technique scales better than the other one and exhibits significant advantages in matchmaking as well as insertion, deletion, and update time against the state-of-the-art work and the other proposed technique. However, it is only able to deal with unary-constrained non-functional service specifications. This disadvantage is solved by the "Subsumes matching" technique which optimizes the non-functional service matchmaking time but pays the price of higher insertion, deletion, and update time. Both techniques can be distributed and incorporated in scalable, distributed service discovery mechanisms. Future work will concentrate on not only optimizing the insertion, deletion, and update time of the "Subsumes matching"

technique but also on developing scalable and distributed, functional and non-functional service discovery mechanisms that incorporate these two novel non-functional service matchmaking techniques. In this way, if the latter mechanisms are exploited by a service broker, then the vision of the Internet of Services will come closer to its realization.

2.5 Cost Reduction Through SLA-driven Self-Management

Contributing partners: INRIA

Status: Presented at the 9th IEEE European Conference on Web Services (ECOWS'11), 14th–16th September, 2011, Lugano, Switzerland.

2.5.1 Background

The Service-Oriented Computing (SOC) promotes the conception of service-based applications on top of loosely-coupled services from distinct providers [54]. The relationship between services are defined by means of electronic contracts which are often represented by Service-Level Agreements (SLAs). The SLA defines the obligations of both provider and customer services which includes the quality that should be provided by the provider [4, 6].

Because service execution environment is distributed, service providers often take advantage of distributed computing infrastructures as clouds [1, 53, 44] and grids [17, 24]. On the one hand, clouds provide resources on-demand along with an accounting model which allows service providers to pay for resources according to their usage. On the other hand, grids provide further programming abstractions useful for managing service instances on distributed resources.

2.5.2 Problem Statement

Cost reduction is a common concern among service providers once it positively contributes to increase their profit. However, it is a challenge to reduce costs while maintaining conformance to SLAs on top of distributed infrastructures. Indeed, just assuring the quality properties described in the SLA is a complex task owing to service load fluctuations and unpredictable faults. Moreover, it is not trivial by far to understand and translate high-level quality metrics and map them to system configuration in order to properly configure service instance to meet the agreed QoS (Quality of Service). In addition to these issues, the fact of dealing with a distributed environment makes harder to build a solution for the stated problem.

2.5.3 Contribution Relevance

Most of current work which tackles SLA management in SOC does not specify actual low-level mechanisms which ensure QoS properties. These approaches typically ensure QoS by replacing services by other services which probably are more suitable for guaranteeing the QoS. Moreover, this service replacement solution is placed in the service composition level where composite services are composed based on simpler services [23, 19]. However, such approaches do not address how basic, atomic services guarantee QoS properties. Additionally, further approaches have addressed SLA management in the context of large-scale distributed applications, such as e-science applications deployed on grids, or multi-tier enterprise applications deployed on clusters [18, 5, 25]. Even though these latter approaches considers SLA aspects, they do not address meeting the business objectives of service providers which involve profit aspects.

2.5.4 Contribution Summary

This work relies on the Qu4DS (Quality Assurance for Distributed Services) framework for managing SLA by minimizing provider costs. In this context, costs refer to infrastructure usage and fine payments owing to SLA violations. In order to reduce costs on infrastructure usage, Qu4DS shares a pool of booked resources among distinct SLA contracts. Regarding fine payments, Qu4DS prevents SLA violations by providing QoS assurance mechanisms which handle dynamic events, such as resource shortages and execution faults. The QoS assurance mechanisms are guided by configurable strategies which attempt to minimize fine payments by choosing the most suitable request to abort in case of resource shortages.

Furthermore, Qu4DS integrates a rich set of QoS management mechanisms which address SLA lifecycle, i.e., from SLA template creation to service termination. Qu4DS builds on a simple interface which is compatible with modern grid and cloud IaaS interfaces. In order to map resource-level configurations to QoS metrics, Qu4DS specifically translates QoS metrics to the right resource requirements able to meet the agreed QoS. Based on the translate resource requirements, resources are acquired on-demand through a common cloud IaaS interface. Then, service instances and requests are dealt with by managing jobs on the booked resources through a grid interface which is based on the Simple Grid API (SAGA) [17]. Moreover, in order to accommodate fluctuating service loads and unpredictable faults, Qu4DS uses dynamic adaptation techniques based on multiple interacting control loops. These control loops are configurable with distinct adaptation policies which allows to extend the applicability of the framework.

2.5.5 Contribution Evaluation

The *flac2ogg* service provider was implemented by using Qu4DS in order to evaluated Qu4DS effectiveness in performing SLA-driven self-management. The flac2ogg is a service provider which encodes audio files by compressing FLAC [70] files to the OGG [71] format. It concerns a Master/Worker application which delegates to Qu4DS the task of managing the execution of its workers during request treatment. Additionally, Qu4DS also assists the flac2ogg provider by managing contract negotiation, translating QoS to resource requirements, booking resources and deploying flac2ogg instances with the right resource configuration. In addition, Qu4DS treats customer requests by reacting to resource shortages or job faults and delays which may compromise to satisfy the agreed QoS.

The Qu4DS framework has been evaluated through experiments on top of Grid5000 [9]. A total of forty-three resources were used for a customer demand composed by fifteen customers with distinct types of SLA. Qu4DS showed to be efficient by reacting to faults and thus successfully treating customer requests. During resource shortages, Qu4DS managed to choose the more suitable request to treat by aiming at increasing the provider profit. These actions were favorable for increasing the provider thus approximating the general provider profit to the ideal profit. More details about this experiment can be found in [15].

2.5.6 Conclusions

This work presented the Qu4DS framework which supports to build services on top of distributed infrastructures, such as IaaS clouds. The framework provides SLA management features which enables service provider to negotiate, instantiate and deliver their service in accordance to quality properties held by the SLA terms. More specifically, Qu4DS offer an automatic support for service execution management by taking into account SLA prices, fines, and infrastructure costs. Therefore, the approach proposed by this work fills the gap between higher-level service objectives and the runtime environment by providing actual mechanisms which manage service execution on distributed infrastructures.

Finally, Qu4DS design relies on configurable and extensible control loops which allow to increase the framework applicability to further application domains, workload characteristics and adaptation objectives. A case study was implemented and evaluates on Grid5000 which demonstrates the framework

effectiveness in increasing the provider profit while maintaining SLA compliance in dynamic and distributed environments.

2.6 Autonomic SLA-aware Service Virtualization for Distributed Systems

Contributing partners: SZTAKI, Vienna University of Technology (TUW)

Status: Presented at the 19th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP'11)

Keywords: Cloud Computing, SLA-negotiation, Service Brokering, On-demand deployment

2.6.1 Background

Cloud Computing citeBuyya2009 builds on the latest achievements of diverse research areas, such as Grid Computing, Service-oriented computing, business processes and virtualization. Both Grids and Service Based Applications (SBAs) already provide solutions for executing complex user tasks, but they are still lacking non-functional guarantees. The newly emerging demands of users and researchers call for expanding service models with business-oriented utilization (agreement handling) and support for human-provided and computation-intensive services. Providing guarantees in the form of Service Level Agreements (SLAs) are highly studied in Grid Computing. Nevertheless in Clouds, infrastructures are also represented as a service that are not only used but also installed, deployed or replicated with the help of virtualization.

2.6.2 Problem Statement

In Cloud infrastructures services are not only used but also installed, deployed or replicated with the help of virtualization. These services can also appear in complex business processes, which further complicates the fulfillment of SLAs in Clouds. For example, due to changing components, workload and external conditions, hardware and software failures, already established SLAs may be violated. Frequent user interactions with the system during SLA negotiation and service executions (which are usually necessary in case of failures), might turn out to be an obstacle for the success of Cloud Computing. Thus, there is demand for the development of SLA-aware Cloud middleware, and application of appropriate strategies for autonomic SLA attainment. Despite cloud computing's business-orientation, the applicability of Service-level agreements in the Cloud middleware is rarely studied, yet [76]. Most of the existing work address provision of SLA guarantees to the consumer and not necessarily the SLA-based management of loosely coupled Cloud infrastructure. In such systems it is hard to localize where the failures have happen exactly, what the reason is for the failure and which reaction should be taken to solve the problem. Such systems are implemented in a proprietary way, making it almost impossible to exchange the components (e.g. use another version of the broker). Autonomic Computing is one of the candidate technologies for the implementation of SLA attainment strategies. Autonomic systems require high-level guidance from humans and decide, which steps need to be done to keep the system stable [29]. Such systems constantly adapt themselves to changing environmental conditions. Similar to biological systems (e.g. human body) autonomic systems maintain their state and adjust operations considering changing components, workload, external conditions, hardware and software failures.

2.6.3 Contribution Relevance

In this contribution we have introduced a novel architecture considering resource provision using a virtualization approach and combining it with the business-oriented utilization used for the SLA agreement

handling. The solution can be used to autonomously manage diverse service infrastructures for on-demand service provision based on SLAs. We also exemplified how autonomic behaviour appears in the architecture in order to cope with changing user requirements and on demand failure handling.

2.6.4 Contribution Summary

The main contributions of this paper include: (i) the presentation of the novel loosely coupled architecture for the *SLA-based Service virtualization* and on-demand resource provision, (ii) description of the architecture including *meta-negotiation, meta-brokering, brokering and automatic service deployment* with respect to its autonomic behaviour, and (iii) the *validation* of the SSV architecture with a biochemical case study in a Cloud simulation environment.

2.6.5 Contribution Evaluation

In order to evaluate our proposed SSV solution, we use a typical biochemical application as a case study called TINKER Conformer Generator application using molecular modeling for drug development, which was gridified and tested on production Grids. The application generates conformers by unconstrained molecular dynamics at high temperature to overcome conformational bias then finishes each conformer by simulated annealing and energy minimization to obtain reliable structures. Its aim is to obtain conformation ensembles to be evaluated by multivariate statistical modeling. This use case can be regarded as a special, corresponding case of the S-Cube EHEALTH-BG-03 scenario called "Easier Planning of Examinations and Treatments". For the evaluation, we have created a general simulation environment, in which all stages of service execution in the SSV architecture can be simulated and coordinated in both Cloud and Grid environments. SLA parameters have been predefined for each task of the application, and have been evaluated resource selection at runtime. From the achieved results we can clearly see that the simulated SSV architecture using different distributed environments overperforms the former solution using only Grid resources. Comparing the different deployment strategies we can see that on demand deployment introduces some overhead, but service duplication can enhance the performance and help to avoid SLA violations with additional virtual machine deployment costs.

2.6.6 Conclusions

The presented general, conceptual SSV architecture is built on three main components: the Meta-Negotiator responsible for agreement negotiations, the Meta-Broker for selecting the proper execution environment, and the Automatic Service Deployer for service virtualization and on-demand deployment. We have also discussed how the principles of autonomic computing are incorporated to the SSV architecture to cope with the error-prone virtualization environments. The proposed service virtualization architecture is validated in a simulation environment based on CloudSim, using a general biochemical application as a case study. The evaluation results clearly fulfil the expected utilization gains compared to a less heterogeneous Grid solution.

Chapter 3

Conclusions

3.1 Outlook and Future Research Challenges

This deliverable introduces a novel service runtime infrastructure, which will incorporate an active and QoS-aware registry and client components. This infrastructure ensures SLA compliance and suggests services as well as ad-hoc processes. The presented results related to elements and aspects of infrastructure and higher level mechanisms for specifying SLAs in regard of QoS requirements. These mechanisms are targeting the objectives of the 2.3 package.

To summarize the significant contributions: TUW studied the combination of service metadata, Quality of Service, service querying, dynamic binding and service mediation. Further, in their work the cost of adaptation on SLA violations is discussed. A third aspect is the establishment of some QoS in a service-based crowd environment. UoC propose two alternative techniques for improving the non-functional service matchmaking time. INRIA's studies consider a SLAs management which satisfies the customers requirements and also their own business objectives, such as maximizing profits. Finally, SZTAKI introduces an SLA-aware Service Virtualization architecture that provides non-functional guarantees in the form of SLAs and consists of a three-layered infrastructure including agreement negotiation, service brokering and on demand deployment.

The deliverable is a collection of scientific papers, either published in conference proceedings, journals, or very recent work still under review, and organized along the research directions of WP-JRA-2.3. The papers all have been peer reviewed which ensures that the papers represent significant contributions to service-based system research and they demonstrate a final progress in the WP. The positioning of the papers within the adaptation framework, their relationship to the WP-JRA-2.3 research goals and vision and to other research WPs are exposed in Section 1.5. A more in detail discussion follows in Chapter 2. Here each contribution gives background information, states the problem statement, and the contribution relevance. This is followed by a contribution summary, an evaluation description and a conclusion with future ideas.

Despite the fact that the present deliverable is the last in the JRA-2.3 Workpackage series, all contributors agree, that future work will not only continue on individual tracks, but also consider collaborations that have certainly also been promoted in the context of SCube. The work on Self-* Service Infrastructures and Service Discovery Support will continue also after the project's end with the hot topic of Cloud Computing rising in the service infrastructure community. The trend can also be recognized from some of the contributions in this deliverable.

Bibliography

[1] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/, April 2011.

[2] Amazon Mechnical Turk. http://www.mturk.com, last access March 2011.

[3] Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. Web Services Agreement Specification (WS-Agreement). Technical report, Open Grid Forum (OGF), 2006. http://www.gridforum.org/documents/GFD.107.pdf, Last Visited: 2011-07-19.

[4] Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. Web Services Agreement Specification (WS-Agreement). Technical report, Global Grid Forum, 2007.

[5] Siegfried Benkner and Gerhard Engelbrecht. A Generic QoS Infrastructure for Grid Web Services. *Advanced International Conference on Telecommunications / Internet and Web Applications and Services, International Conference on*, 0:141, 2006.

[6] Philip Bianco, Grace A. Lewis, and Paulo Merson. Service Level Agreements in Service-Oriented Architecture Environments. Technical Report CMU/SEI-2008-TN-021, Software Engineering Institute of The Carnegie Mellon University, http://www.sei.cmu.edu/reports/08tn021.pdf, 2008.

[7] David Bodoff, Mordechai Ben-Menachem, and Patrick C.K. Hung. Web Metadata Standards: Observations and Prescriptions. *IEEE Software*, 22(1):78–85, 2005.

[8] D.C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75, 2008.

[9] F. Cappello, E. Caron, M. Dayde, F. Desprez, Y. Jegou, P. Primet, E. Jeannot, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, B. Quetier, and O. Richard. Grid'5000: A large scale and highly reconfigurable grid experimental testbed. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, GRID '05, pages 99–106, Washington, DC, USA, 2005. IEEE Computer Society.

[10] Owen Cliffe and Dimitris Andreou. Service Matchmaking Framework. Public Deliverable D5.2a, Alive EU Project Consortium, 10 September 2009. Available at: http://www.ist-alive.eu/index.php?option=com_docman&task=doc_download&gid=28&Itemid=49.

[11] Antonio Ruiz Cortés, Octavio Martín-Díaz, Amador Durán Toro, and Miguel Toro. Improving the Automatic Procurement of Web Services Using Constraint Programming. *Int. J. Cooperative Inf. Syst.*, 14(4):439–468, 2005.

[12] Asit Dan, Doug Davis, Robert Kearney, Alexander Keller, Richard P. King, Dietmar Kuebler, Heiko Ludwig, Mike Polan, Mike Spreitzer, and Alaa Youssef. Web Services on Demand: WSLA-Driven Automated Management. *IBM Systems Journal*, 43:136–158, January 2004.

[13] Seema Degwekar, Stanley Y. W. Su, and Herman Lam. Constraint Specification and Processing in Web Services Publication and Discovery. In *ICWS*, pages 210–217, 2004.

[14] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Mass collaboration systems on the World Wide Web. *Communications of the ACM*. to appear.

[15] André Lage Freitas, Nikos Parlavantzas, and Jean-Louis Pazat. Cost Reduction Through SLA-driven Self-Management. In *In Proceedings of The 9th IEEE European Conference on Web Services (ECOWS'11)*, September 2011.

[16] Andreas Friesen and Michael Altenhofen. Matching Composed Semantic Web Services at Publishing Time. In *Proceedings of the IEEE International Conference on E-Commerce Technology Workshops*, pages 64–70, Munich,Germany, 2005. IEEE Computer Society.

[17] Tom Goodale, Shantenu Jha, Hartmut Kaiser, Thilo Kielmann, Pascal Kleijer, Gregor von Laszewski, Craig Lee, Andre Merzky, Hrabri Rajic, and John Shalf. Saga: A simple api for grid applications - high-level application programming on the grid. *Computational Methods in Science and Technology: special issue "Grid Applications: New Challenges for Computational Methods"*, SC05:8(2), November 2005.

[18] Peer Hasselmeyer, Bastian Koller, Lutz Schubert, and Philipp Wieder. Towards SLA-Supported Resource Management. In *HPCC '06: Proceedings of the 2006 International Conference on High Performance Computing and Communications*, pages 743–752. Springer, 2006.

[19] Julia Hielscher, Andreas Metzger, and Raman Kazhamiakin. Taxonomy of Adaptation Principles and Mechanisms. Technical Report Deliverable # CD-JRA-1.2.2, S-CUBE Consortium, 2009.

[20] Waldemar Hummer, Philipp Leitner, and Schahram Dustdar. SEPL – A Domain-Specific Language and Execution Environment for Protocols of Stateful Web Services. *Distributed and Parallel Databases*, 29:277–307, August 2011.

[21] C. Hwang and K. Yoon. Multiple Criteria Decision Making. *Lecture Notes in Economics and Mathematical Systems*, 1981.

[22] Panagiotis G. Ipeirotis. Analyzing the Amazon Mechanical Turk Marketplace. *SSRN eLibrary*, 17(2):16–21, 2010.

[23] Florian Irmert, Thomas Fischer, and Klaus Meyer-Wegener. Runtime adaptation in a service-oriented component model. In *SEAMS '08: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems*, pages 97–104, New York, NY, USA, 2008. ACM.

[24] Shantenu Jha, Andre Merzky, and Geoffrey Fox. Using Clouds to Provide Grids with Higher Levels of Abstraction and Explicit Support for Usage Modes. *Concurrency and Computation: Practice & Experience*, 21:1087–1108, 2009.

[25] Jose Antonio Parejo and Pablo Fernandez and Antonio Ruiz-Cortés and José María García. SLAWs: Towards a Conceptual Architecture for SLA Enforcement. In *Services, IEEE Congress on*, volume 0, pages 322–328. IEEE Computer Society, 2008.

[26] Lukasz Juszczyk, Hong-Linh Truong, and Schahram Dustdar. GENESIS - A Framework for Automatic Generation and Steering of Testbeds of Complex Web Services. In *Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'08)*. IEEE Computer Society, 2008.

[27] 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+)*, pages 395–404, 2009.

[28] Alexander Keller and Heiko Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal on Network and Systems Management*, 11:57–81, March 2003.

[29] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[30] A. Kertész, G. Kecskemeti, and I. Brandic. Autonomic sla-aware service virtualization for distributed systems. In *In proceedings of the 19th Euromicro International Conference on Parallel, Distributed and Network-Based Computing*, 2011.

[31] Matthias Klusch, Benedikt Fries, and Katia Sycara. OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(2):121 – 133, 2009.

[32] Jacek Kopecký, Tomas Vitvar, Carine Bournez, and Joel Farrell. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing*, 11:60–67, November 2007.

[33] K. Kritikos and B. Pernici. Initial concepts for specifying end-to-end quality characteristics and negotiating slas. Technical report, 2009.

[34] Kyriakos Kritikos. QoS-based Web Service Description and Discovery. PhD Thesis, Computer Science Department, University of Crete, Heraklion, Greece, December 2008.

[35] Kyriakos Kritikos, Fabio Paternò, and Dimitris Plexousakis. Towards Identifying Services to Realize the Functionality of Interactive Applications based on User Task Models. *ACM Transactions on Interactive Intelligent Systems*, 2011. under review.

[36] Kyriakos Kritikos and Dimitris Plexousakis. Mixed-Integer Programming for QoS-Based Web Service Matchmaking. *IEEE Trans. Serv. Comput.*, 2(2):122–139, 2009.

[37] André Lage Freitas, Nikos Parlavantzas, and Jean-Louis Pazat. Cost Reduction Through SLA-driven Self-Management. In *European Conference on Web Services (ECOWS)*, Lugano, Switzerland, September 2011. The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant greement 215483 (S-CUBE). Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

[38] P. Leitner. Requirements for service infrastructures in dynamic environments and evaluation of existing service registries. Technical report, 2009.

[39] P Leitner, B Wetzstein, F Rosenberg, A Michlmayr, S Dustdar, and F 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)*, pages 176–186, Berlin, Heidelberg, 2009. Springer-Verlag.

[40] Philipp Leitner, Waldemar Hummer, and Schahram Dustdar. Cost-Based Optimization of Service Compositions. *IEEE Transactions on Services Computing (TSC)*, 2011. To appear.

[41] Philipp Leitner, Florian Rosenberg, and Schahram Dustdar. Daios – Efficient Dynamic Web Service Invocation. *IEEE Internet Computing*, 13(3):30–38, 2009.

[42] 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 International Conference on Service-Oriented Computing (ICSOC'10)*. Springer, 2010.

[43] Tammo van Lessen, Jörg Nitzsche, and Frank Leymann. Formalising Message Exchange Patterns using BPEL Light. In *Proceedings of the 2008 IEEE International Conference on Services Computing (SCC'08)*, pages 353–360, Washington, DC, USA, 2008. IEEE Computer Society.

[44] Lillard, Terrence V. and Garrison, Clint P. and Schiller, Craig A. and Steele, James. *The Future of Cloud Computing*, pages 319–339. Elsevier, 2010.

[45] Yutu Liu, Anne H. H. Ngu, and Liangzhao Zeng. Qos computation and policing in dynamic web service selection. In *WWW (Alternate Track Papers & Posters)*, pages 66–73, 2004.

[46] Juval Löwy. *Programming WCF Services*. O'Reilly, 2007.

[47] Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng. Semantic Web Services. *IEEE Intelligent Systems*, 16(2), 2001.

[48] Daniel A. Menascé. QoS Issues in Web Services. *IEEE Internet Computing*, 6(6):72–75, 2002.

[49] Anton Michlmayr, Florian Rosenberg, Philipp Leitner, and Schahram Dustdar. Service Provenance in QoS-Aware Web Service Runtimes. In *Proceedings of the 7th International Conference on Web Services (ICWS'09)*. IEEE Computer Society, 2009.

[50] Anton Michlmayr, Florian Rosenberg, Philipp Leitner, and Schahram Dustdar. End-to-End Support for QoS-Aware Service Selection, Binding, and Mediation in VRESCo. *IEEE Transactions on Services Computing*, 3:193–205, July 2010.

[51] Anton Michlmayr, Florian Rosenberg, Christian Platzer, Martin Treiber, and Schahram Dustdar. Towards Recovering the Broken SOA Triangle – A Software Engineering Perspective. In *Proceedings of the 2nd International Workshop on Service Oriented Software Engineering (IW-SOSWE'07), co-located with ESEC/FSE'07*. ACM, 2007.

[52] Sonia Ben Mokhtar, Davy Preuveneers, Nikolaos Georgantas, Valérie Issarny, and Yolande Berbers. EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. *J. Syst. Softw.*, 81(5):785–808, 2008.

[53] Nurmi, Daniel and Wolski, Rich and Grzegorczyk, Chris and Obertelli, Graziano and Soman, Sunil and Youseff, Lamia and Zagorodnov, Dmitrii. The Eucalyptus Open-Source Cloud-Computing System. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 124–131, 2009.

[54] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. Service-Oriented Computing: State of the Art and Research Challenges. *Computer*, 40:38–45, 2007.

[55] M. Parkin and A. Metzger. Initial set of principles, techniques and methodologies for assuring end-to-end quality and monitoring of slas. Technical report, 2010.

[56] Pierluigi Plebani and Barbara Pernici. URBE: Web Service Retrieval Based on Similarity Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1629–1642, 2009.

[57] H. Psaier, L. Juszczyk, F. Skopik, D. Schall, and S. Dustdar. Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems. In *SASO*, pages 164–174. IEEE, 2010.

[58] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubén Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Christoph Bussler, and Dieter Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 2005.

[59] Florian Rosenberg, Predrag Celikovic, Anton Michlmayr, Philipp Leitner, and Schahram Dustdar. An End-to-End Approach for QoS-Aware Service Composition. In *Proceedings of the 13th International Enterprise Computing Conference (EDOC'09)*. IEEE Computer Society, 2009.

[60] Florian Rosenberg, Philipp Leitner, Anton Michlmayr, and Schahram Dustdar. Integrated Metadata Support for Web Service Runtimes. In *Proceedings of the Middleware for Web Services Workshop (MWS'08)*, pages 361–368, Washington, DC, USA, 2008. IEEE Computer Society.

[61] Florian Rosenberg, Christian Platzer, and Schahram Dustdar. Bootstrapping Performance and Dependability Attributes of Web Services. In *Proceedings of the IEEE International Conference on Web Services (ICWS'06)*, pages 205–212, Washington, DC, USA, 2006. IEEE Computer Society.

[62] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006.

[63] O. Sammodi and A. Metzger. Integrated principles, techniques and methodologies for specifying end-to-end quality and negotiation slas and for assuring end –to-end quality provision and sla conformance (incl. proactiveness). Technical report, 2011.

[64] B. Satzger, H. Psaier, D. Schall, and S. Dustdar. Stimulating skill evolution in market-based crowdsourcing. In *9th International Conference on Business Process Management (BPM)*, volume 6896 of *Lecture Notes in Computer Science*, pages 66–82. Springer, 2011.

[65] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley, New York, N.Y, USA, 1986.

[66] F. Silvestri. Knowledge extraction of service usage. Technical report, 2010.

[67] James Skene, D. Davide Lamanna, and Wolfgang Emmerich. Precise Service Level Agreements. In *Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*, pages 179–188, Washington, DC, USA, 2004. IEEE Computer Society.

[68] Florian Skopik, Daniel Schall, and Schahram Dustdar. Trust-Based Adaptation in Complex Service-Oriented Systems. In *Proceedings of the 2010 15th IEEE International Conference on Engineering of Complex Computer Systems*, pages 31–40, Washington, DC, USA, 2010. IEEE Computer Society.

[69] M. Stollberg, M. Hepp, and J. Hoffmann. A Caching Mechanism for Semantic Web Service Discovery. In *ICWS*, 2007.

[70] The FLAC project. Free Lossless Audio Codec (FLAC). http://flac.sourceforge.net/, 2011.

[71] The Xith Open Source Community. Ogg Vorbis Audio Format. http://www.vorbis.com/, October 2011.

[72] M. Vukovic. Crowdsourcing for Enterprises. In *Proceedings of the 2009 Congress on Services*, pages 686–692. IEEE Computer Society, 2009.

[73] Yao Wang and Julita Vassileva. Toward Trust and Reputation Based Web Service Selection: A Survey. *International Transactions on Systems Science and Applications (ITSSA)*, 3(2), 2007.

[74] Branimir Wetzstein, Philipp Leitner, Florian Rosenberg, Schahram Dustdar, and Frank Leymann. Identifying Influential Factors of Business Process Performance Using Dependency Analysis. *Enterprise Information Systems*, 4(3):1–8, July 2010.

[75] World Wide Web Consortium (W3C). Web Services Description Language (WSDL) Version 2.0 Part 0: Primer - W3C Candidate Recommendation 27 March 2006, 2006. http://www.w3.org/TR/2006/CR-wsdl20-primer-20060327/, Last Visited: 2011-07-19.

[76] C. A. Yfoulis and A. Gounaris. Honoring slas on cloud computing services: a control perspective. In *Proceedings of the European Control Conference*, 2009.

[77] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang. QoS-Aware Middleware for Web Services Composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, May 2004.

[78] Chen Zhou, Liang-Tien Chia, and Bu-Sung Lee. DAML-QoS Ontology for Web Services. In *ICWS*, page 472, San Diego, CA, USA, 2004. IEEE Computer Society.