RC24341 (W0709-010) September 5, 2007
Computer Science

IBM Research Report

Proceedings of the IBM PhD Student Symposium at ICSOC 2007

Ed. by

Tudor Dumitras (Carnegie Mellon University, USA)
Andreas Hanemann (Leibniz Supercomputing Center, Germany)
Benedikt Kratz (Tilburg University, The Netherlands)
Jyotishman Pathak (Iowa State University, USA)

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Preface

The IBM Ph.D. Student Symposium at ICSOC provides a forum where doctoral students conducting research in Service-Oriented Computing (SoC) have the opportunity to present their on-going dissertation work to a group of well-known experts in the field and receive feedback from them. Each presentation is organized as a mock thesis-defense, with a committee of 4 mentors providing extensive feedback and advice for completing a successful Ph.D. thesis. This format is similar to the one adopted by the doctoral symposia associated with ICSE, OOPSLA, ECOOP, Middleware and ISWC.

The closing session of the symposium is a panel discussion where the roles are reversed: the mentors answer the students' questions about research careers in industry and academia. The symposium agenda also contains a keynote speech, delivered by Dr. Paolo Traverso, addressing the "hot topics" and new research challenges in SoC. Dr. Traverso is director of research at IRST, Italy, where he leads a division working on software and services, knowledge management, and embedded systems. He was Program Committee Chair at ICSOC 2004 and General Chair at ICSOC 2005.

This year we have received 17 submissions from 11 countries and 4 continents: 11 from Europe, 3 from North America, 2 from Asia and 1 from South America. As the goal of the symposium is to provide constructive feedback to the authors of both accepted and rejected papers, each paper was reviewed by three Program Committee members (we did not use external reviewers). The submissions were evaluated according to five criteria: quality of research, breadth of background knowledge, relevance to SoC, presentation and project maturity. The program committee has finally selected 7 papers with different maturity levels, guided by the belief that the students presenting their work in the symposium learn from the feedback of the mentors, but also from each other, and that Ph.D. students who are almost ready to graduate set a good example for their peers who are just starting. The papers included in these proceedings address, from both technical and business perspectives, a range of topics including service discovery, composition, interoperability and the establishment of service agreements.

September 2007

Tudor Dumitraş
Andreas Hanemann
Benedikt Kratz
Jyotishman Pathak

Co-Organizers of the IBM Ph.D. Student Symposium

Organization and Committees

The IBM Ph.D. Student Symposium is held in conjunction with the 5th International Conference on Service Oriented Computing (ICSOC 2007). It is organized as a whole-day event on September 17, 2007 at the Technical University of Vienna (Vienna, Austria). The homepage of the symposium is at: http://infolab.uvt.nl/phd-icsoc07/.

Mentoring Committee

Willem-Jan van den Heuvel (Tilburg University, The Netherlands)
Priya Narasimhan (Carnegie Mellon University, USA)
Paolo Traverso (IRST, Italy)
Petr Tůma (Charles University, Czech Republic)

Program Committee

Claudio Bartolini (HP Labs, USA)
Rik Eshuis (Technical University of Eindhoven, The Netherlands)
Willem-Jan van den Heuvel (Tilburg University, The Netherlands)
Alexander Keller (IBM Research, USA)
Priya Narasimhan (Carnegie Mellon University, USA)
Elisabetta di Nitto (Politecnico di Milano, Italy)
Daniela Roşu (IBM Research, USA)
Jennifer M. Schopf (Argonne National Lab, USA)
George Spanoudakis (City University, United Kingdom)
Petr Tůma (Charles University, Czech Republic)
John Wilkes (HP Labs, USA)

Organizing Committee

Tudor Dumitraş (Carnegie Mellon University, USA)
Andreas Hanemann (Leibniz Supercomputing Center, Germany)
Benedikt Kratz (Tilburg University, The Netherlands)
Jyotishman Pathak (Iowa State University, USA)

Sponsoring Institutions

IBM Research, USA

Table of Contents

Exploiting Discovered Web Services in Service-centric Systems Development

Konstantinos Zachos

Centre for HCI Design, School of Informatics
City University, London, UK
kzachos@soi.city.ac.uk

Abstract. Service-centric systems pose new challenges and opportunities for requirements processes and techniques. This paper reports new techniques that enable service discovery during early requirements processes and exploit discovered services to enhance requirements specifications. The paper describes the algorithm for discovering services from requirements expressed using structured natural language. The paper also reports a first evaluation of the utility of the environment that implements this algorithm when improving the specification of requirements with retrieved services.

1. Introduction

Web and software services are operations that users access via the internet through a well-defined interface independent of where the service is executed [14]. Service-centric systems integrate software services from different providers seamlessly into applications that discover, compose and monitor these services.

One consequence of service-centric systems is that requirements processes might change due to the availability of services. Discovering candidate services can enable analysts to increase the completeness of system requirements based on available service features. However, for this to happen, new tools and techniques are needed to form service queries from incomplete requirements specifications. In this paper, new tools and methods for discovering services to use to make requirements specifications more complete are described.

2. Motivation and Problem Statement

From the perspective of software engineering, requirements engineering (RE) is the first activity of the software process. At this stage the goal is to establish what services are required from the system and the constraints on the system's operation and development [6]. During early requirements processes a requirements team seeks to establish system boundaries, acquire and discover requirements, and explore dependencies both with adjacent systems and between actors in the system being specified. At this stage, requirements specifications are typically informal and incomplete — an opportunity to discover services early in the process to explore capabilities and features for a new application. In this research we attempt to exploit this opportunity, i.e. how service descriptions that are retrieved in response to service queries can refine and decompose the original requirements that gave rise to the query. In this paper, we propose a method that uses semantic matching strategies to discover services that are semantically similar to the deployed system specification.

Furthermore, recently more and more research has been invested into the idea of inventing rather than merely elicit requirements – a missing step in the requirements process [7,10,11]. The RE process has been revealed as highly creative involving cycles of insight-driven re-conceptualisation of the problem space [10], especially in early phases of requirements processes (e.g. [8]). Most service consumers, on their own, are not inventors. Requirements analysts are the people whose skills and position allow and encourage creativity [11]. The emergence of new systems means that stakeholders will increasingly create and invent ideas that they express as requirements. For instance, studies such as Maiden et al. [8] have shown that people can exploit analogies to reuse requirements if they are given support to recognise, understand and transfer the analogies. In this paper, we propose a solution that uses analogical reasoning techniques to encourage the creation and invention of new requirements. Rather than rely on expert presentations of domain knowledge, the proposed method will discover and retrieve services in analogical domains to the current service consumer application, to support creative thinking about requirements for that application.

Therefore, the primary objectives of the research are

- to design and implement a *requirements-based service discovery model* to discover services from service registries that are semantically similar to the deployed system specification; and
- to design and implement a *creativity-driven service discovery model* that enhances requirements-based service discovery model in order to expand the search space and discover descriptions of candidate services in analogical domains to the current service consumer application.

3. Related Work

There is a growing body of work on service discovery based on techniques from software reuse. Most existing approaches (e.g. [2]) use semantically richer service description frameworks such as OWL-S [1] to deliver flexible matching between service queries and descriptions. However, these are inappropriate for early requirements processes, when queries cannot be expressed in such precise, formal and complete terms. Wang et al. [15] report a web service discovery technique that combines WordNet with structure-matching on the structure of the WSDL service specification. As in our solution, WordNet is used to expand queries with semantically similar words. However, this technique is limited because only formal representations in WSDL are considered.

In requirements research there has been little reported work on service discovery. Schmid et al. [12] report a requirements-led process to enable run-time service discovery, but no tool support is described. Elsewhere Esmaeilsabzali et al. [4] report new models for requirements-based service discovery that assume formal expression of system operations. Our main innovation is to expand service queries to handle requirements expressed in natural language and compatible with established processes. In addition, current RE creativity techniques and approaches (e.g. [10]) do not take service-centric systems engineering into account as a means to support creative thinking about requirements for a system. In this research, we will exploit analogical reasoning techniques to encourage the creation and invention of new requirements.

4. Service Discovery using Natural Language Requirements

RE is often divided into early and late requirements processes. Figure 1 depicts this characteristic – the divergence from and convergence to established requirements. During early requirements, services are used to encourage divergence activities to challenge system boundaries and assumptions, and discover new requirements. The presented solution uses service descriptions to support tasks that encourage creative thinking by discovering services that are analogical to services that will be implemented in the deployed system, to encourage exploratory creative thinking about requirements. During late requirements processes services can be used to support convergent activities such as decomposing and refining specifications of requirements, and restructuring the requirements to enable more effective service monitoring. This thesis focuses on the divergence activities during early requirements processes.

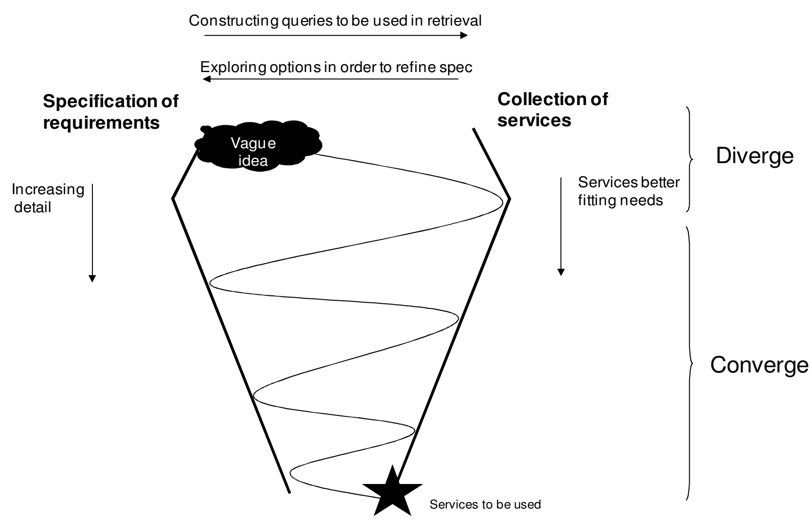

Figure 1. SeCSE's Requirements Process

The research gathered throughout the course of this thesis and the results of the evaluation will be used to determine whether the findings lead to an acceptance of the following two main hypotheses:

(1) During early requirements processes, requirements-based queries can discover services that an analyst can use to generate new, and revise existing requirements appropriate in the domain, and
(2) During early requirements processes, an analyst can use services discovered from requirements-based queries to invent new requirements, i.e. requirements that are novel and appropriate in the domain.

4.1. Requirements-based Service Discovery

Current literature suggests that an overwhelming majority of requirements specifications are written in natural language (NL), often augmented or enhanced by information in other notations, such as formulae, and diagrams [3]. Since requirements specifications form the basis of the whole process, it is desirable for requirements specifications to be as correct, complete, and unambiguous as possible. However, specifying requirements in NL has some major drawbacks, namely the inherent ambiguity and incompleteness of NL.

The requirements-based service discovery engine called EDDiE (Expansion and Disambiguation Discovery Engine) overcomes these common characteristics of NL requirement specifications by introducing two capabilities. These capabilities are designed to generate queries that will discover services using requirements that are ambiguous and incomplete. Consider the requirement for a car's route planning system: *the system shall provide the driver with directions to a chosen destination by the most direct route*. It is incomplete because it does not state what the directions are and what direction information is needed. It is also ambiguous because it does not define what is the sense of the "most direct" route. There are several possible meanings of *direct*. Does the analyst mean *direct in spatial dimensions; proceeding without deviation or* interruption, or does s/he mean having *no intervening persons or agents*?

To handle incompleteness and ambiguity when discovering services we extend query expansion techniques previously only applied to WSDL service specifications [15] to incomplete statements of requirement to generate more complete service queries. And we apply term disambiguation techniques from information retrieval [13] to ambiguous statements of requirement to generate unambiguous service queries. The claimed innovation is to import research from related disciplines and extend it to handle problems specific to RE and service discovery.

The algorithm has the 4 key components shown in Figure 2. The WordNet on-line lexicon [9] fulfils an important role for three of the algorithm's components. In the first the service request is divided into sentences, then tokenized and part-of-speech tagged and modified to include each term's morphological root (e.g. *driving* to *drive*, and *drivers* to *driver*). Secondly, the algorithm applies procedures to disambiguate each term by defining its correct sense and tagging it with that sense by iteratively using context knowledge from the project glossary, requirements analyst and other terms in the service query (e.g. defining a *driver* to be a *vehicle* rather than a *type of golf club*). Thirdly, the algorithm expands each term with other terms that have similar meaning according to the tagged sense, to increase the likelihood of a match with a service description (e.g. the term *driver* is synonymous with the term *motorist* which is also then included in the query). In the fourth component the algorithm matches all expanded and sense-tagged query terms to a similar set of terms that describe each candidate service in the service registry. Query matching is in 2 steps: (i) XQuery text-searching functions to discover an initial set of services descriptions that satisfy global search constraints; (ii) traditional vector-space model information retrieval, enhanced with WordNet, to further refine and assess the quality of the candidate service set.

Services identified using EDDiE's matching engine against requirements information may lead directly to the identification of new requirements as developers discover services which are ready for use in the domain of interest, but which they were previously unaware of.

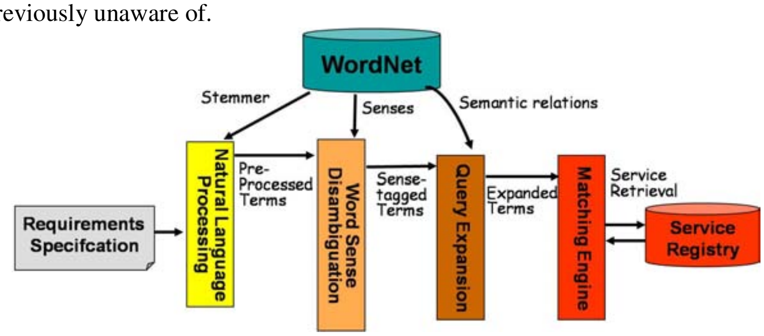

Figure 2. Requirements-based service discovery algorithm

4.2. Creativity-driven Service Discovery

As mentioned in section 2, RE is increasingly a creative process in which stakeholders and designers work together to create ideas for new systems, then express these ideas as requirements that envision these new ideas. This section describes a model called AnTiQue (Analogy Tracker in Service Queries) which uses analogical reasoning techniques to encourage the creation and invention of new requirements.

The main function of AnTiQue's service discovery algorithm is to discover descriptions of candidate services in analogical domains to the current service consumer application. AnTiQue's similarity model is based on predicate logic and the Structure Mapping Theory (SMT) [5] where NL text is presented as predicates, i.e. prepositional networks of nodes (objects) and edges (predicate values). We distinguish two kinds of predicates. *Attributional* predicates used to state properties of objects expressed in the form *PredicateValue(Object)* and *relational* predicates used to express relations between objects expressed in the form *PredicateValue(Object1, Object2)*; for instance, "the car is red" becomes *red(car)* and "the driver drives the car" becomes *drive(driver, car)*. An *analogy* is a comparison in which relational predicates, but few or no attributional predicates, can be mapped from source to target. For instance, intended inferences of the comparison "constructing a building is like developing software" mainly concern the relational structure e.g., "a programmer designs software components (*design(programmer, software component)*), just as the architect designs building plans (*design(architect, building plan)*)" but none or a few concerning attributes. On the other hand, a *literal similarity* statement is a comparison in which a large number of predicates are mapped from source to target including **both** attributional and relational predicates.

AnTiQue's algorithm implements this notion of analogy by extending functionalities provided by EDDiE.

5. Evaluation and Future Work

We evaluated EDDiE with experienced practitioners who had extensive experience with automotive applications. Rather than investigate traditional measures of precision and recall properties of the EDDiE algorithm itself we investigated the utility of algorithm in a requirements workshop. More specifically we explored whether discovered services were sufficient to trigger specification of requirements that had not been specified prior to service discovery. Our assumption underlying this strategy was that high levels of precision and recall were not essential for service discovery – service specifications with lower similarity scores might still enable analysts to discover new requirements. The evaluation revealed that experienced practitioners used retrieved services with a range of similarity scores to generate new requirements that were later ranked as more novel than requirements discovered using traditional use case walkthrough techniques. This positive outcome supports our fundamental claim that candidate services can enable analysts to increase the completeness of requirements.

Based on these results, we will evaluate the use of AnTiQue to explore the question whether analysts can discover new requirements from discovered analogical services that are novel and appropriate in the domain.

References

1. Ankolenkar A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D.L., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne T.R., and Sycara, K., 2002, "DAML-S: Web Service Description for the Semantic Web", The First International Semantic Web Conference (ISWC), Italy.
2. Benatallah B., Hacid, M., Rey, C., Toumani, F., 2003, 'Request Rewriting-Based Web Service Discovery', International Semantic Web Conference.
3. Berry D.M., Kamsties, E. & Krieger, M., 2003, 'From contract drafting to software specification: Linguistic sources of ambiguity, a handbook', Tech. rep., School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.
4. Esmaeilsabzali, S., Day N. & Mavadatt F., 2005, 'Specifying Search Queries for Web Service Discovery', Proceedings SOCCER (Service-Oriented Computing: Consequences for Engineering Requirements) Workshop, at RE'05 Conference, Paris, August 2005.
5. Gentner D., 1983, 'Structure-mapping: A theoretical framework for analogy', Cognitive Science, 7.
6. Kauppinen, M., Vartiainen, M., Kontio, J., Kujala, S. & Sulonen, R., 2004, 'Implementing requirements engineering processes throughout organizations: success factors and challenges', Information & Software Technology, 46, 937–953.
7. Maiden N. & Jones S., 2005, 'Creativity in the design of complex systems', Workshop on Complexity in Design and Engineering, University of Glasgow.
8. Maiden N.A.M., Gizikis A. & Robertson S., 2004, 'Provoking creativity: Imagine what your requirements could be like', IEEE Software, 21, 68–75.
9. Miller, K., 1993, 'Introduction to WordNet: an On-line Lexical Database'.
10. Nguyen L. & Swatman P., 2005, 'Promoting and supporting the creative and insight-driven re process using design rationale', Working Paper, 15, university of South Australia, Adelaide.
11. Robertson J., 2005, 'Requirements analysts must also be inventors', IEEE Software, 22, 48–50.
12. Schmid K., Eisenbarth M., Grund M., 2005, 'From Requirements Engineering to Knowledge Engineering: Challenges in Adaptive Systems', Proceedings SOCCER (Service-Oriented Computing: Consequences for Engineering Requirements) Workshop, at RE'05 Conference, Paris, August 2005.
13. Schutze, H., Pedersen, J., 1995, 'Information retrieval based on word senses', Proceedings of the Symposium on Document Analysis and Information Retrieval 4, 161–175.
14. Tetlow, P., Pan, J., Oberle, D., Wallace, E., Uschold, M., Kendall, E., 2005, 'Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Engineering', W3C.
15. Wang Y., and Stroulia, E., 2003, 'Semantic Structure Matching for Assessing Web-Service Similarity', First International Conference on Service Oriented Computing, Trento, Italy.

Ontology and Behaviour aware Service Discovery

Sara Corfini

Department of Computer Science, University of Pisa, Italy
corfini@di.unipi.it

Abstract. A major challenge for Service-oriented Computing is how to discover and compose services to build complex applications. In this paper we present a methodology for the discovery of compositions of services that accounts for the need of (1) composing several services to satisfy a client request, and (2) guaranteeing useful behavioural properties of the composite service. The methodology makes use of functional and behavioural information contained in OWL-S service advertisements.

1 Introduction: Context and Features

Service-oriented Computing (SoC) [1] is emerging as a new promising computing paradigm that centers on the notion of service as the fundamental element for developing distributed software applications. Roughly, a Web service is any piece of software that makes itself available over the Internet. A Web service is identified by a URI, it is universally accessible by means of standard protocols (WSDL, UDDI, SOAP), and it self-describes its functionalities by exposing a public interface. *Web service discovery* is a key issue of SoC, as it allows developers to find and re-use existing services to rapidly build complex applications.

The standard service description language (WSDL) provides services with purely syntactic descriptions, including neither behavioural information on the possible interaction among services, nor semantics (viz., ontology-based) information to describe the functionality of services. Yet, both behavioural and semantic information may be necessary, for example, to satisfy complex queries that require to compose the functionalities offered by different services, as well as to automate the processes of service discovery and composition.

During the last years, various proposals have been put forward to feature more expressive service descriptions that include both semantics and behaviour information about services. One of the major efforts in this direction is OWL-S [2], a high-level ontology-based language for describing services. In particular, OWL-S service descriptions include a list of semantically annotated functional attributes of services (the *service profile*), and a declaration of the interaction behaviour of services (the so-called *process model*).

The objective of this paper is to present a behaviour-aware, ontology-based methodology for discovering OWL-S described services. In particular, the methodology employs semantic (viz., ontology-based) information to select available services that can be exploited to satisfy a given query, and it employs behaviour information to suitably compose such services to achieve the desired result. The main features of the methodology can be summarised as follows:

1. *Ability of addressing functional and behavioural queries* – that is, respectively, queries specifying the functional attributes of the desired service, and queries also requiring a specific behaviour of the service to be found. In particular, in case of a behavioural query, the discovery methodology – besides satisfying the query functional requirements – guarantees that the returned service features the desired behaviour.
2. *Composition-oriented matching* – that is, the capability of discovering service compositions. When no single service can satisfy the client query, the methodology checks whether the query can be fulfilled by a suitable composition of services.
3. *Ontology-based matching* – that is, the ability of "crossing" different ontologies and performing flexible matching automatically. Given that different services are typically described in terms of different ontologies, the discovery methodology determines relationships between concepts defined in separate ontologies, so to establish functional dependencies among services.
4. *Automatic service discovery* – that is, the ability of processing both functional and behavioural client queries not requiring human interaction.
5. *Efficiency* – which is obviously an important objective of any service discovery mechanism. In particular, in order to avoid affecting efficiency further, the discovery methodology pre-computes off-line (i.e., before query answering time) all the query-independent tasks (e.g., to determine the functional dependencies among services as well as the relationships among ontologies).

It is worth mentioning that preliminary results of our research towards an automated behaviour-aware discovery have been published in [3, 4] (that mainly focus on feature 2). Moreover, the current version of the discovery methodology has been recently published in [5].

The rest of the paper is organised as follows. We present the behaviour-aware, ontology-based discovery methodology in Section 2. We draw some concluding remarks (concerning related work, open issues and future work) in Section 3.

2 A Methodology for a Behaviour-aware Discovery

In this Section, we present a methodology for discovering compositions of semantic Web services which takes into account both *semantic* and *behavioural* information advertised in the OWL-S service descriptions. In particular, we employ "semantics", namely all those ontological information regarding the functional attributes (i.e., inputs and outputs) of services, to select services with respect to "what they really do", and we employ "behaviour", namely, information concerning the order with which messages can be received or sent by each service, to guarantee some useful properties of selected services. Before presenting the discovery methodology, we introduce the data structures and formalisms we employ to synthesise service descriptions.

The internal representation of services
As briefly mentioned in the Introduction, the complete behaviour of a service is described by the OWL-S process model, which may include conditional and

iterative constructs. Hence, a service may behave in different ways and feature different functionalities. We say that a service may have different *profiles*, each of them requiring/providing different inputs/outputs. Hence, as one may expect, we represent each service with two distinct items: a *set of profiles*, to summarise the different sets of functional attributes employed by each profile of the service, and a *Petri net*, to model the whole service behaviour.

More precisely, a profile S_n represents a dependency between the set of the inputs and the set of the outputs employed by the specific behaviour n of a service S. Service profiles are collected into a *hypergraph*, whose nodes correspond to the functional attributes of the service profiles, and whose hyperedges represent relationships among such attributes. It is worth observing that each node v of the hypergraph, that is, each functional attribute, is associated with a concept, which is defined in one of the ontologies referred by the service employing v. The hypergraph also includes equivalent and sub-concept relationships among nodes, viz., among ontology concepts. (A formal definition of the hypergraph and the algorithms for its construction can be found in [6].)

The complete interaction behaviour of a service is represented by an OCPR net. OCPR nets (for Open Consume-Produce-Read nets) [7] are a simple variant of the standard Condition/Event Petri nets, designed to naturally model the behaviour of services. Briefly, an OCPR net is equipped with two disjoint sets of places, namely, *control* places and *data* places (that can be produced but not consumed), to properly model the control flow and the data flow of a Web service, and with an interface, which establishes those data places that can interact with the external environment. We formally defined OCPR nets in [7], where a mapping from OWL-S process models to OCPR nets is also presented.

Discovering compositions of services

So far, we have introduced the internal representation of services that we use to store them in a local repository. We can now propose a complete composition-oriented methodology for discovering services. The methodology takes as input the so-called *behavioural queries*, that is, queries specifying both the inputs and outputs, as well as the expected behaviour of the service to be found. A behavioural query, for example, can be expressed in terms of the OWL-S process model describing the desired service. The set of the functional attributes of the query can be easily retrieved by its OWL-S process model, which can be in turn suitably translated into an OCPR net [7]. Hence, we can assume that a query consists of two parts: a couple $\langle I, O \rangle$ (viz., a *functional* query) and an OCPR net, respectively describing the set of the inputs and outputs, and the behaviour of the service to be found. The discovery methodology we are going to propose consists of two main phases: a *functional analysis* and a *behavioural analysis*.

Functional analysis.

The aim of this phase is to filter services (more precisely, *profiles*) with respect to their functional attributes only. In particular, for each set of services S returned by the functional analysis the following conditions hold: (1) all the query outputs are provided by the services in S, (2) all the inputs of the services in S are provided by the query (or they can be produced by some service in S).

As previously described, we summarise functional information of the services stored in the repository in a hypergraph. The functional analysis hence consists of a visit of the hypergraph. It is worth noting that by exploring *profiles*, we address the discovery of sets of services, as well as by exploring *sub-concept* and *equivalent* relationships we properly reason with (different) ontologies. In particular, the functional analysis explores the hypergraph starting from those nodes corresponding to the query outputs, and it continues by visiting backwards the hyperedges until reaching, if possible, the query inputs. The profile-labelled hyperedges which take part in a hyperpath from the query outputs to the query inputs determine a set of service profiles satisfying the query. A detailed discussion of the algorithm for visiting the hypergraph can be found in [6].

Behavioural analysis.
Each set S of services returned by the functional analysis is then taken as input by the behavioural analysis, which checks whether the services included in S, suitably composed together, are *behaviourally* equivalent to the client query. The behavioural analysis consists of the following two main steps.
1) Constructing the composite service. In the first step the behavioural analysis constructs the OCPR net modelling the parallel composition of the services included in S. To this aim, the behavioural analysis firstly retrieves from the local repository those OCPR nets modelling the complete behaviour of the services in S, and next, it constructs the composite OCPR net $N_{\|}$ by performing the parallel composition of the retrieved nets. In particular, given that the data places which belong to the net interface are the only ones that can interact with the external environment, in order to compose OCPR nets, we have to operate on their interfaces. Hence, after performing the disjoint union of the transitions, data places and control places of the nets modelling the services in S, next, we collapse those data places which are (syntactically and/or semantically) equivalent and which occur in the interfaces of such nets. Moreover, in order to perform the parallel composition of the retrieved nets, we add to $N_{\|}$ the necessary additional transitions and control places, according to the OCPR mapping of the parallel composition (viz., the OWL-S `split+join` construct) given in [7].
2) Analysing the service behaviour. The second step of the behavioural analysis checks whether the composition of those services previously selected during the functional analysis is capable of satisfying the query from a behavioural perspective. Let N_Q denote the net representing the behavioural query. Namely, this step checks whether N_Q and $N_{\|}$ are equivalent, that is, whether they are externally indistinguishable. To this end, we defined in [7] a suitable notion of behavioural equivalence for Web services, which features *weakness*, as it equates structurally different yet externally indistinguishable services; *compositionality*, as it is also a congruence; and *decidability*, as the set of states that an OCPR net can reach is finite. More precisely, a state of an OCPR net is the marking of its observable places. Intuitively speaking, the behavioural analysis checks whether – for every possible interaction with the external environment – each state of N_Q can be reached by $N_{\|}$ (and vice versa). Hence, the behavioural equivalence of N_Q and $N_{\|}$, with respect to the definition given in [7], guarantees that the

set S of services previously returned by the functional analysis also satisfies the behavioural requirements of the client query.

3 Concluding remarks: Open Issues and Future Work

The automated composition-oriented, ontology-based methodology for discovering (semantic) Web services previously presented is the first – at the best of our knowledge – that takes properly into account functional, semantic and behaviour information provided by service descriptions.

Recently, automatic matchmaking of Web services has gained prominent importance and new approaches are frequently introduced. For the lack of space, we briefly discuss hereafter only some of the more closely related approaches. A more extended discussion of related work can be found in [3, 4, 7]. The first effort towards the automation of Web service discovery has been put forward by some of the authors of OWL-S in [8]. Their algorithm performs a functionality matching between service requests and service advertisements described as DAML-S (the predecessor of OWL-S) service profiles. This approach was the first at introducing the notion of an automatic and flexible matching by suitably considering subsumes and plug-in relationships among the ontology-annotated attributes of services and service requests. Yet, the algorithm proposed in [8] does not deal with the ontology crossing problem, that is, it is not able to determine relationships between attributes annotated with concepts of separate ontologies. To this aim, it is worth mentioning the service matchmaking approach presented by Klusch et al. in [9], which employs both logic based reasoning and IR techniques to properly relate concepts of different ontologies.

A common drawback of [8, 9] is that they search for a *single* service capable of satisfying a client query by itself. However, as previously described, composing functionalities of different services may be necessary to satisfy a query. An approach to a composition-oriented discovery is presented by Benatallah et al. in [10], where the matchmaking problem is reduced to a best covering problem in the domain of hypergraph theory. Yet, it is important to stress that none of the mentioned proposals takes into account behavioural aspects of services, namely, [8, 10, 9] are capable of neither solving behavioural queries nor guaranteeing behavioural properties of the returned (compositions of) services.

Behavioural aspects of services are partially taken into account by the approach of Agarwad and Studer, that proposed in [11] a new specification of Web services, based on description login and π-calculus. Their algorithm consider semantic and temporal properties of services, yet, their matchmaking approach is limited to a single service discovery.

The methodology briefly introduced in this paper hence advances the state-of-the-art approaches by featuring a behaviour-aware discovery capable of addressing important issues such as the discovery of service compositions that feature a desired behaviour. However, there are two main open issues. The first concerns the *efficiency* and *scalability* of the discovery methodology, indeed, functional and behavioural analysis require exponential time. In this setting, indexing and/or ranking techniques (as search engines do for Web pages) may play a central role to sensibly improve the efficiency of the discovery methodology. Yet, the problem in this case seems to be a bit harder than for text indexing due to the nonmonotonicity of caching issues, indeed, adding/removing one parameter in a query may radically change the results associated to a query. The second open issue concerns the *generality* of the discovery methodology, i.e., the ability of going *beyond* OWL-S. In order to avoid interning the methodology to OWL-S services, it is important to extend it in order to cope with other services, e.g., with WS-BPEL services. For instance, a possible solution is to translate BPEL processes into OWL-S services, by employing the BPEL*2*OWL-S translator developed by Aslam et al. in [12]. Yet, there is still a prominent problem, namely, the lack of ontological information in WS-BPEL descriptions. Hence, when a provider adds a WS-BPEL process into the system, firstly, the WS-BPEL process is translated into a "rough" OWL-S service, and secondly, the provider is asked to complete the OWL-S description, by annotating the service parameters with ontology concepts. (e.g., by employing some friendly ontology editor, such as Protégé (`http://protege.stanford.edu`).

References

1. Papazoglou, M.P., Georgakopoulos, D.: Service-Oriented Computing. Communications of the ACM **46**(10) (2003) 24–28
2. OWL-S Coalition: OWL-S: Semantic Markup for Web Service (2006) `http://www.ai.sri.com/daml/services/owl-s/1.2/overview/`.
3. Brogi, A., Corfini, S., Popescu, R.: Semantic-based Composition-oriented Discovery of Web services. ACM Transaction on Internet Technology **8**(4) (2008) *To appear.*
4. Brogi, A., Corfini, S.: Behaviour-aware discovery of Web service compositions. International Journal of Web Services Research **4**(3) (2007) 1–25
5. Benigni, F., Brogi, A., Corfini, S.: Discovering Service Compositions That Feature a Desired Behaviour. In Krämer, B., Lin, K., Narasimhan, P., eds.: ICSOC 2007. *LNCS 4749*, Springer (2007) 56–68
6. Brogi, A., Corfini, S., Aldana, J., Navas, I.: A Prototype fot Discovering Compositions of Semantic Web Services. In Tumarello, G., Bouquet, P., Signore, O., eds.: Proc. of the 3^{rd} Italian Semantic Web Workshop. (2006)
7. Bonchi, F., Brogi, A., Corfini, S., Gadducci, F.: A behavioural congruence for Web services. In Arbab, F., Sarjani, M., eds.: Fundamentals of Software Engineering, *LNCS* (To appear), Springer (2007)
8. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic Matchmaking of Web Services Capabilities. In Horrocks, I., Hendler, J., eds.: 1^{st} Int. Conf. on The Semantic Web, *LNCS* 2342, Springer (2002) 333–347
9. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with OWLS-MX. In: AAMAS'06, ACM (2006) 915–922
10. Benatallah, B., Hacid, M.S., Léger, A., Rey, C., Toumani, F.: On automating Web services discovery. VLDB J. **14**(1) (2005) 84–96
11. Agarwal, S., Studer, R.: Automatic Matchmaking of Web Services. In: IEEE Int. Conference on Web Services, IEEE Computer Society (2006) 45–54
12. Aslam, M.A., Auer, S., Shen, J., Herrmann, M.: Expressing Business Process Models as OWL-S Ontologies. In Eder, J., Dustdar, S., eds.: Business Process Management Workshops. *LNCS* 4103, Springer (2006) 400–415

Type-based Validation and Management of Business Service Interoperability

Toni Ruokolainen

Dept. of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 UNIVERSITY OF HELSINKI, FINLAND
Toni.Ruokolainen@cs.Helsinki.FI

Abstract. To attain interoperable business service collaborations, the nature and concepts of service-oriented inter-enterprise environments must be carefully analysed and formalized. Moreover, development tools and runtime infrastructure services are needed for establishing a sustainable business service ecosystem that provides interoperable business service delivery. The research proposal introduced in this paper addresses these issues by developing metamodels and methods for validating and maintaining business service interoperability in the context of inter-enterprise collaborations.

1 Introduction

Modern networked enterprises require flexibility and openness from collaboration facilities to tolerate changes in the ever-changing technology and business domains, and to gain a competitive edge. However, heterogeneity, autonomy, and dynamism inherent in inter-enterprise computing environments present severe interoperability problems when loosely coupled collaborations should be established. To attain interoperable business service collaborations, the nature and concepts of service-oriented inter-enterprise environments must be carefully analysed and formalized. More over, development tools and runtime infrastructure services are needed for establishing a sustainable business service ecosystem that provides interoperable business service delivery.

The research proposal introduced in this paper addresses the interoperability problem by developing methods for validating and maintaining business service interoperability in the context of inter-enterprise collaborations. The research is thematically located within the areas of service-oriented computing [1] and modern middleware research [2], model-driven engineering (MDE) [3], as well as formal methods [4–6]. The contributions of the research are focussed mainly on the areas of service-oriented computing and integrated formal methods while MDE is adopted as a supporting framework and software engineering discipline.

The structure of this paper is as follows. Section 2 discusses the related research areas and technologies, and identifies challenges to be addressed. The research theme and approach is introduced in Section 3. Finally, the expected results and contributions are reflected upon in Section 4.

2 Background and Significance

The openness of collaborations and agility of organizations stressed by modern electronic business necessitates very flexible interaction relationships between individual enterprise information systems. The degree of flexibility can be increased using the *service-oriented computing* (SOC) approach which is claimed to attain a very loosely coupled model of distributed computing and to provide means for the "open service markets" [1]. From the author's perspective, the SOC field still lacks support for three essential elements that can truly bring the ideal of open service markets closer. These missing elements and challenges to address are *1)* a feasible trading mechanism for interoperable business service delivery, *2)* mechanisms for attaining loosely coupled service relationships, and *3)* a methodology for service-oriented software engineering (SOSE).

The research is motivated by the preceding challenges. The approach taken is based on a notion of *service typing* [7] that gives formal rigour to such concepts as business service behaviour, interoperability and substitutability. Metamodels are provided for service definition and description with theoretical foundations for verifying interoperability between service definitions, and validating the correspondence of a service description (business service offer) with respect to the claimed service definition (service type). Such validation and verification functionalities play an important role when considering service trading in truly open environments. In addition to a feasible service trading infrastructure such concepts as separation of concerns, runtime binding of business services, and dynamic configurability of service capabilities are needed to attain loose coupling in service-oriented systems. These concepts are addressed by the research and reflected in the metamodels, as for example in the form of metamodels for non-functional aspects [8]. Finally, the notion of service typing and corresponding toolchain is believed to fertilize a SOSE methodology; this hypothesis and the feasibility of the methodology is to be validated during the research.

3 Research Theme and Approach

The overall theme of my research can be characterised as type-based validation and management of business service interoperability in the context of inter-enterprise collaborations. The research follows a constructive research methodology with phases of conceptualization, formalization, artefact construction, and evaluation of the approach. The elementary contents of the constructive research phases are discussed in the following.

3.1 Conceptualizing service-oriented collaborations

In the conceptualization phase, the target concepts from the domain of inter-enterprise collaborative systems are identified and formalized into a set of metamodels. The resulting modeling framework follows the principles of MDE [3] and comprises a hierarchy that is illustrated in Figure 1. The highest layer consists of the MOF constructs [9] applicable for defining metamodels. The *collaborative systems metamodel* provides target concepts for specific kinds of collaborative systems. The set of concepts include notions for collaboration, interaction, service-oriented computing etc. The concepts defined at the *federated service communities metamodel* describe such notions as service types [7], business network models [2], and electronic contracts. In addition for defining modelling constructs for creating the knowledge required, the metamodels prescribe the essential facilities for an abstract service-oriented computing platform.

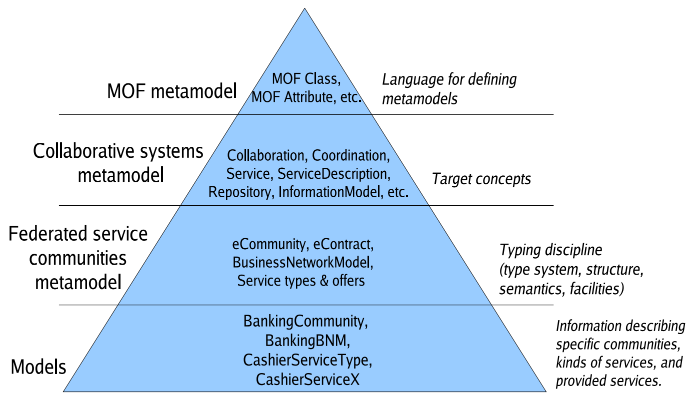

Fig. 1. A four-level modelling hierarchy for describing federated service communities.

Preliminary versions of the metamodels have been introduced in [10]. The complete versions of the metamodels are defined in the author's Licentiate's thesis. The metamodels are illustrated using UML class diagrams. The Object Constraint Language (OCL) is used for further refining the semantics of the metamodels by defining refinement relationships and inter-dependencies between different metamodels and their concepts. Currently the set of metamodels contains over 15 metamodels (packages) with over 80 classes. The metamodels are concretely developed over the Eclipse Modeling Framework (EMF) [1].

3.2 Formalizing business service interoperability and collaborations

The formalization involves construction of a typing discipline that involves three essential type structures, namely 1) service types, 2) binding types, and 3) collaboration types. While service types consider the behaviour of individual business services, the two latter prescribe consistency rules for bilateral and multilateral service interactions (i.e. service compositions and choreographies), respectively. The abstract syntax for all the preceding type structures is given by the federated service communities metamodel depicted in Figure 1.

A *service type* is an abstract description of service capabilities, behaviour and structure [7]. It is a unit of design and composition which represents a bilateral business

[1] http://www.eclipse.org/emf

service interface. Service typing is a scheme of behavioural typing where service descriptions are considered as terms that have to be well-typed with respect to corresponding service types. The service typing discipline is based on session typing [5], pi-calculus [4], and research conducted on the areas of XML-typing [6] and process algebras with XML-typing such as the PiDuce [11]. For service typing purposes the original session typing concept is extended with constructs for handling XML-like structured data (see e.g. [6]) and the tagged braching constructs (see [5]) that are reminiscents of the RPC-paradigm are replaced with constructs compatible with the message passing semantics of SOC. Service typing as the foundation for business service interoperability has been discussed in [7] with an initial discussion about the type management infrastructure and its applicability during the business service development process.

A *binding type* characterizes a bilateral business service connection. It provides the semantics for the connection by defining business protocols used at the connection endpoints and, when necessary, a set of transformations to establish compatibility between the endpoints. Behavioural compatibility between business protocols and thus consistency of the binding type is validated by methods based on the session typing discipline [5]. In addition to business protocols, a binding type may define non-functional properties for the connection (e.g. "*secured interaction*" or "*non-repudiable service*") and business protocol transformations. A preliminary formalization of the non-functional properties has already been conducted [8]. The business protocol transformations may involve mappings between business document structures or even transformations between behavioural patterns.

Service composition and choreographies are managed by so-called *collaboration types* that define the well-formedness and consistency rules for such service collaborations. There are two major research challenges to be addressed: 1) giving type theoretic interpretations for service collaborations, and 2) developing the consistency criteria and their validation methods for the collaboration types. Collaboration types for service compositions and choreographies are constructed as compositions between a set of service types and corresponding coordination structures. A coordination structure formalizes the inter-dependencies between service types; feasibility of so-called event structures [12] as the logical framework for collaboration types will be evaluated. The consistency criteria for collaboration types includes the absence of circular relationships between causal dependencies and the behaviour induced by the coordination structure and service type behaviours, since such a relationship would result in a deadlock. Moreover, satisfaction of coordination obligations prescribed in coordination structures by the service types has to be validated; a static analysis framework, possibly based on the flow logics [13], has to be developed for this purpose.

The semantics of business processes and collaboration choreographies will be founded on pi-calculus [4]-based process calculus. The labelled selection constructs used in the process calculi of session typing disciplines such as described in [5] are replaced with pattern matching constructs. In the process calculus, sessions are created over explicit session creation channels as usual [5]; however, the use of session creation channels shall be governed by a linear usage-typing discipline that provides means for prescribing obligations or prohibitions with respect to the usage of corresponding communication channels. The process calculus must also incorporate XML-based messaging. XML-based messaging has previously been used in process calculi such as the PiDuce [11], but the session typing has not been used in the calculi the author is aware of.

3.3 Constructing the facilities for service-oriented computing

In the constructive part of the research a set of meta-information repositories and software engineering tools are developed. The repositories and tools are based on the metamodels and formalisms developed during the previous research efforts. All of the routine management code is provided by the metamodels, and the code generation facilities provided by the Eclipse environment. Most of the actual work expected in this phase considers the implementation of the verification and static analysis algorithms developed during the formalization phase.

The type management infrastructure is a public and distributed meta-information management system which maintains service type information and relationships between service types. Type management infrastructure consists of public type repositories and name registries. Type repositories implement type checking and type matching functionalities which are needed for interoperation validation during collaboration establishment. Name registries are used for name-based resolution of meta-information such as service types and business network models in the Pilarcos interoperability middleware [2]. The role of name registries is not a foundational part of my research but their existence is necessary for delivering a business service trading environment.

The development tools to be constructed include modelling tools for service types, collaboration types and business network models for example. The development tools utilize the meta-information repositories discussed above. For the purpose of integrating the different development tools and meta-information repositories the Eclipse MDDi framework [2] can be utilized.

4 Expected Results and Contribution

The expected results elaborate the concepts of service-oriented computing and especially the role of formal service definitions as part of service-oriented architectures and software engineering processes. Moreover, research and development of facilities to attain loosely coupled business service collaborations will be strongly represented in this research. The foundations for these are laid by the conceptualization work in the form of metamodels and typing disciplines.

The incredients for a SOSE methodology are addressed by the research proposal. First of all, the service typing discipline provides support for interoperable service discovery and delivery while the collaboration types address service composition issues. Secondly, the semantics of business processes and corresponding type-based interoperability validation procedures can be attached to SOSE design and development tools. Using the results stemming from the previous research areas, a tool-chain and lightweight methodology for SOSE can ultimately be constructed. Consequently, this re-

[2] `http://www.eclipse.org/mddi`

search also contributes to the work on integrating formal methods in software engineering processes.

Feasibility of the service typing discipline, metamodels and corresponding development and SOA facilities are validated using case studies. The feasibility of the service type concept must be validated from a practical perspective: is it valuable for business service developers, and are the type checking and other analysis procedures implementable in the development tools and meta-information repositories?

References

1. Papazoglou, M.P., Georgakopoulos, D.: Special issue on Service-Oriented Computing. Commun. ACM **46** (2003)
2. Kutvonen, L., Ruokolainen, T., Metso, J.: Interoperability middleware for federated business services in web-Pilarcos. International Journal in Enterprise Information Systems **3** (2007) 1–21
3. Schmidt, D.C.: Model-Driven Engineering. Computer **39** (2006) 25–31
4. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, part I/II. Journal of Information and Computation **100** (1992) 1–77
5. Vallecillo, A., Vasconcelos, V.T., Ravara, A.: Typing the behavior of objects and components using session types. Electronic Notes in Theoretical Computer Science **68** (2003) Presented at FOCLASA'02.
6. Hosoya, H., Pierce, B.C.: XDuce: A statically typed XML processing language. ACM Trans. Inter. Tech. **3** (2003) 117–148
7. Ruokolainen, T., Kutvonen, L.: Service Typing in Collaborative Systems. In Doumeingts, G., Müller, J., Morel, G., Vallespir, B., eds.: Enterprise Interoperability: New Challenges and Approaches, Springer (2007) 343–354
8. Ruokolainen, T., Kutvonen, L.: Characterizing Non-Functional Aspects of Services in Federated Service Communities. Manuscript. (2007)
9. Object Management Group: Meta Object Facility (MOF) Core Specification. 2.0 edn. (2006) OMG Available Specification – formal/06-01-01.
10. Ruokolainen, T., Kutvonen, L., Metso, J.: Ontology for Federated Management of Business Networks. In: 2nd International Workshop on Interoperability Solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems. (2007)
11. Brown, A., Laneve, C., Meredith, G.: PiDuce: A process calculus with native XML datatypes. In: 2nd International Workshop on Web Services and Formal Methods (WS-FM 2005), Versailles, France (2005)
12. Winskel, G., Nielsen, M.: Models for concurrency. In Abramsky, S., Gabbay, D.M., Maibaum, T.S.E., eds.: Semantic Modelling. Volume 4 of Handbook of Logic in Computer Science. Clarendon Press (1995)
13. Nielson, H.R., Nielson, F.: Flow Logic: a multi-paradigmatic approach to static analysis. In T. Mogensen et al., ed.: The Essence of Computation: Complexity, Analysis, Transformation. Essays dedicated to Neil D. Jones. Volume 2566 of Lecture Notes in Computer Science. Springer Verlag (2002) 223–244

Business Driven Strategy for Service-Oriented Architecture: A Systems Dynamics Approach

Jae Choi

Sheldon B. Lubar School of Business,
University of Wisconsin-Milwaukee,
3202 N. Maryland Ave. Milwaukee, WI 53201, U.S.A.
choij@uwm.edu

Abstract. With firms facing intense rivalry, globalization, and time-to-market pressures, organizational agility and Information Systems (IS) agility assume greater importance. However, despite the emergence of Service-Oriented Architecture (SOA) as a prominent agility-enhancing technology, the fundamental question of how new technologies such as SOA can shape a firm's competitive advantage over time has yet to be resolved. The complexity associated with the dynamic interactions among the many factors in the business environment, organizational and IS agility, technology adoption, and Information Technology (IT) human resource management, contributes to a lack of clear strategy for implementing new technologies like SOA to achieve recognizable business value. This research attempts to examine the impact of potential strategies surrounding SOA under changing conditions, and provide guidance for managerial decisions, using a systems dynamics approach. The model and accompanying simulation can serve as a practical decision support system for the strategic decisions of implementing SOA.

Keywords: Service-Oriented Architecture, Systems Dynamics, Information Systems Agility, IT-Business Alignment

1. Introduction

A recent survey indicates that SOA has become a key imperative for a majority of business and a dominant technology issue in IT market [8], since it can serve as a vehicle to address IT-business alignment, thereby, facilitating a firm's agility. For an organization, the business value associated with SOA is coupled with the preferred level of agility which varies depending upon a variety of internal and external factors. Further, the dynamic interactions among factors affecting the business value of SOA, like IT-Business alignment and agility are also influenced by other aspects of business, e.g., technology adoption and IT human resource management, among others. The complexity and the dynamic nature of the business value of SOA domain make it difficult to assess the implications of strategic decisions surrounding SOA, and whether more appropriate decisions could have been made by an organization given prevailing organizational, environmental, and contextual factors. The inability to effectively assess the consequences of SOA related strategic decisions in an a-priori manner leaves managers to speculate if the decisions made were indeed the appropriate ones. A model that captures the complexities, dynamics, and a range of such decisions could contribute to the knowledge base for adoption and effective implementation of SOA. In this study, we intend to develop a computational model of the mechanics underlying the strategic and business value aspect of SOA, representing this as dynamic interactions among the multi-faceted factors that shape the focal problem domain.

2. IS Agility, Alignment, Business Value, and SOA

Researchers identify IT as being one of the core enabler for organizational agility [9]. However, IT can be also a potential constraint or inhibitor of organizational agility, particularly where systems are monolithic and not compatible. Without addressing such potentially negative impact of IT, the likelihood of any significant improvement in organizational agility directly attributable to IT is questionable. Prior research has identified key knowledge bases, and IT architectural frameworks such as Component-Based Development (CBD), Web Services, and SOA as potential springboards for improving IS agility.

IT-business alignment leads to superior organizational performance, improving the strategic and business value of IT. However, the task of maintaining close IT-business alignment under rapidly changing business and technological environments is challenging especially when IT infrastructure is monolithic, inflexible, and incompatible, thereby entailing lower IS agility. As their environment evolves, firms need to constantly adjust alignment by reconfiguring technological structure. Though agile IS plays a crucial role in shaping strategic and business value by facilitating IT-business alignment, the relationship between IS agility and the strategic/ business value of IT is not clearly explicated. Several factors tend to complicate this relationship. First, IT contribution to business value is typically dependent upon multiple other factors, including firms' industry characteristics [2] and the type of competition [4], among others. Second, strategic decisions for a firm's key activities e.g., IT human resource management, technology adoption, or the implementation of SOA and other agility-enhancing technologies, greatly affect the relationship between IS agility and a firm's performance. Our primary aim is to develop a computational model that captures this spectrum of environmental and organizational conditions in a manner that permits dynamic analysis of the phenomena across multiple time periods. The need for time-based analysis, coupled with complex interactions among underlying constructs suggests the use of simulation-based approaches for modeling.

3. Modeling Methodology: Systems Dynamics

A host of modeling techniques are available for studying dynamic relationships between organizational constructs addressing SOA adoption and business value. This dissertation adopts the view that the relationships between organizational, environmental, and contextual factors cannot be modeled in simple linear terms, and that scrutiny of performance over time is necessary to understand the underlying phenomena. As a result, many of the exact modeling techniques like optimization are rendered inapplicable. Simulation, using a dynamic approach appears more meaningful. Systems dynamics, a form of simulation that permits modeling of behavior among related constructs as well as across successive time periods appears particularly relevant in this context. It has been already proved useful for IS studies, including business planning for network services [5], offshore outsourcing [6], and software project management [1], among others. A primary advantage of systems dynamics is that it facilitates the representation of both quantifiable variable and hard-to-measure variables [5].

Assembling the model requires the initial identification of stocks and converters that form the basis for the formal model. Next, these stocks and converters need to be related to each other, both within and across time periods. The resulting set of interlinked feedback loops reflects the processes of the simulated world. Calibrating and validating the model forms the next step. Application to a variety of organizational and environmental conditions rounds out its development. When applying the model, our emphasis is on analyzing behavior patterns, trends, and how quickly a variable achieves stability, rather than on absolute values generated through the simulation. Recalibration of the model equations will permit focus on the actual numbers.

4. The Business-Technology Value Model

Prior research in the area has addressed several parts of the adoption-value phenomenon. However, a comprehensive model that links SOA adoption to business value is clearly lacking. This research seeks to address this gap. It posits that SOA adoption, IT human resources management, and agility are major determinants of business value that is attributable to IT. Figure 1 presents a high level view of this notion, decomposing the model into these three distinct areas. In actual fact, the model is much more detailed. For example, the skill management subsystem alone contains more than 30 causal links. For illustrative purpose we describe this subsystem in more detail.

IT staffing decisions are typically driven by the skills needed and those possessed by the available workforce. The workforce can be augmented through long term strategies like recruitment and short term strategies like the acquisition of contract workers, consulting arrangements, or outsourcing. It is depleted through turnover, manifest as resignations, retirements, and layoffs. Completion of temporary contracts will also reduce the available workforce. The level of available skill sources is driven by the magnitude of skill gap. However, the choice of skill acquisition is less a function of skill gap magnitude, than is shaped by organizational and environmental factors, such as resource availability and problem urgency. Low levels of urgency and resource availability suggest acquisition through inexpensive means, favoring training as the preferred alternative. Cases involving high levels of problem urgency will suggest rapid acquisition, favoring contracting. Recruitment becomes the preferred option when the urgency is not critical, and sufficient resources are available. After considerable analysis of different situations involving urgency and resource availability, a grid search of the two-dimensional space was performed, generating a partitioning of the space [3]. In its present formulation, the space is divided into three areas based on linear relationships between resource availability and urgency. Preliminary results indicate that environmental uncertainty and resource availability play a key role in the selection of alternative skill acquisition strategies [3].

In figure 1, the value subsystem addresses business agility and IS agility and their impact on business value. This research elects to model two discrepancies as major determinants of agility and hence value – one between business requirements and business processes, and the other addressing business requirements and IT capabilities. Business practice/ processes can become less effective through increased

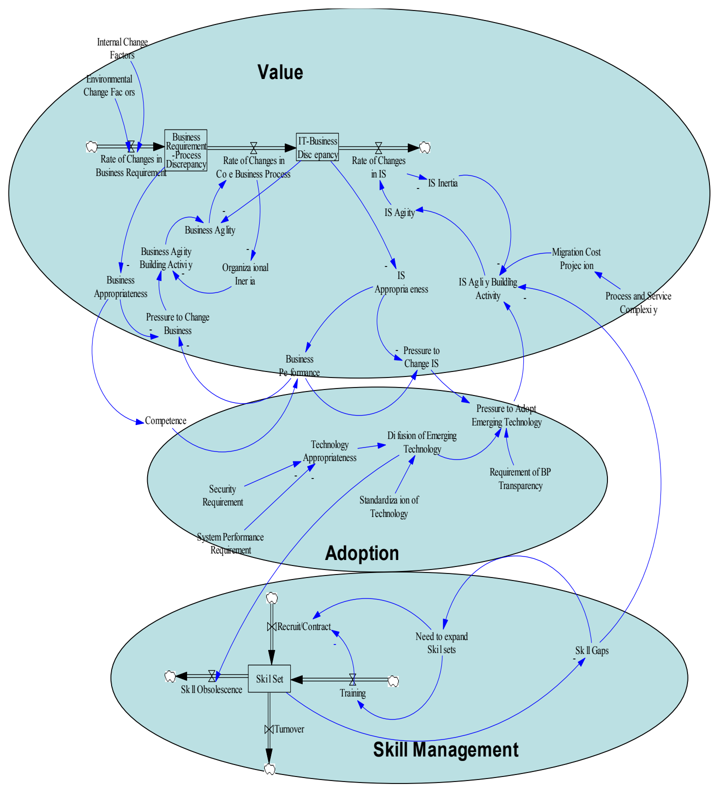

Fig. 1. Business-Technology Value Model

discrepancy between requirement and practice, particularly when the environment changes rapidly. However, depending on the level of business agility, processes can be adjusted, thereby regaining effectiveness. The impact of frequent updates in business process is two-folded. It can make IT less aligned to business, and business agility may eventually degrade as a result of the lower IT-business alignment. In turn, this degraded business agility will hamper the ability of the organization to update its processes quickly, and eventually making the prevailing business practices less relevant. The vicious cycle can be broken by enhanced IS agility or the emergence of less turbulent environment. A narrow IT-Business discrepancy enhances business agility, resulting in more relevant business practices and processes. The model considers SOA implementation as a primary IS agility building activity. A set of inhibitors as well as facilitators of SOA implementation is also embedded in the model to examine a range of decision alternatives and strategies surrounding SOA in relation to the resulting changes in agility, alignment, and business performance. Further, a variety of sourcing options, including in-house implementation, turnkey outsourcing, and a subscription to utility computing service can be easily accommodated within the model framework.

The adoption subsystem is currently the least mature component of the model. It addresses the industry-wide diffusion of SOA and factors that influence a firm's decision of SOA adoption. Industry-wide diffusion is primarily driven by the level of the fit between industry characteristics and the technological attributes of SOA. Together with such normative pressure from industry, a set of firm-specific factors affect adoption decision. We seek to develop a reasonably generic model to accommodate a range of agility-enhancing technologies, but it must be also specific enough to incorporate key details of SOA.

5. Model Validation

This paper employs the design science research methodology [7]. The business technology value model serves as the design artifact in this research. Validation of the model is a critical component of the research. Without the ability to demonstrate that it yields appropriate results, there is little likelihood that the goal of design science, utility can be achieved, hampering its use by decision makers involved in SOA adoption. A multi-pronged approach to model creation and validation will be employed. At the outset, the model was assembled using constructs and relationships that have been identified and verified in prior adoption research. Structural assessment [11] was performed to ensure that the model adequately reflects reality. Causal loops were checked to ensure that they are balancing and not reinforcing. Calibration of the model was performed using parameter assessment [11]. This involved choosing appropriate scales and units for each construct, and assessing whether the numerical relationships reflect the real world. Calibration of the model was performed using real-world data, expert opinion, as well as prior academic and professional literature. Further assessment of the model was performed through simulation under controlled conditions, and an examination of the results to see if they were consistent with expected outcomes. Simulation using the adoption subsystem indicates that the results track favorably with the traditional S-curve for adoption of innovations. Systematic exploration of the problem space will demonstrate how robust the model is. Additional validation can be performed application of the model to real world adoption scenarios, and comparison of predicted results with historical outcomes. Given the applied nature of the research, several different approaches to validation are employed, including the "analysis", "evaluation", and "example" approaches [10].

6. Conclusion and Future work

This paper describes ongoing research to study the dynamics of value crafting process of SOA in relation with multiple subsystems, business activities, and diverse environment. It employs a modeling approach that is based on systems dynamics, permitting managers to explore the implications of alternative decisions. Once calibrated for a specific firm, the model can be used to provide practical decision support to identify suitable strategic decisions underneath the adoption and implementation of SOA. Environmental and organizational conditions can be manipulated to permit systematic what-if analysis. The ability to a-priori assess strategic decisions on SOA, as opposed to relying on lagged post-hoc analysis, affords managers the ability to make more informed decisions. The model permits evaluation of the multi-faceted, dynamically changing, and interactive factors that affect SOA related strategy formulation. It can be calibrated to reflect different organization contexts and emerging features of SOA. The comprehensive full models of value and adoption subsystems will be produced in the next phase. These will then be integrated into a comprehensive model after extensive calibration and validation using real world cases.

References

1. Abdel-Hamid, T. K.: The Dynamics of Software Project Staffing: A System Dynamics Based Simulation Approach. IEEE Transactions on Software Engineering 15 (2) (1989) 109-119
2. Chatfield, A. T., Yetton, P.: Strategic Payoff from EDI as a Function of EDI Embeddedness. Journal of Management Information Systems 16 (4) (2000) 195-224
3. Choi, J., Jain, H., Nazareth, D.: Information Technology Skill Management: A Systems Dynamic Approach. Accepted at the 41st Annual Hawaii International Conference On System Science (HICSS) (2008)
4. Devaraj, S., Kohli, R.: The IT Payoff: Measuring the Business Value of Information Technology Investments. Prentice-Hall Inc., Upper Saddle River, NJ (2002)
5. Dutta, A.: Business Planning for Network Services: A Systems Thinking Approach. Information Systems Research 12 (3) (2001) 260-283
6. Dutta, A., Roy, R.: Offshore Outsourcing: A Dynamic Causal Model of Counteracting Forces. Journal of Management Information Systems 22 (2) (2005) 15-35
7. Hevner, A. R., March, S. T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1) 75-105
8. InfoWorld.: From Pilot to Payoff: Service Oriented Architecture hits its stride. White Paper. InfoWorld and BEA Systems http://www.bea.com/newsletters/arch2arch/2006-04.jsp (2006)
9. Sambamurthy, V., Bharadwaj, A., Grover, V.: Shaping Agility through digital options: Reconceptualizing the role of information technology in Contemporary firms. MIS Quarterly 27(2) (2003) 237-2
10. Shaw, M.: Writing Good Software Engineering Research Papers. Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society (2003) 726-736
11. Sterman, J. D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin McGraw-Hill Company (2000)

A Feature-based Approach to Web Services E-contract Establishment

Marcelo Fantinato

Institute of Computing, University of Campinas, Brazil
mfantina@ic.unicamp.br

Abstract. This Ph.D. research proposes a feature-based approach to be applied in Web services e-contract establishment. It aims at improving the information structure and reuse of e-contracts, including QoS attributes. Features are used to represent possible contract elements in order to drive template instantiation and act, therefore, as a configuration space manager. The approach uses an e-contract meta-model integrating WS-BPEL and WS-Agreement concepts. A tool is being developed and a case study carried out to evaluate the approach feasibility and effectiveness.

Keywords: E-contracts; QoS attributes; information reuse; feature modeling.

1. Introduction

Electronic contracts are used to describe details of the supply and the consumption of e-services within an inter-organizational business process [1], [2]. They may also include levels for QoS attributes agreed between the involved parties [3], [4]. E-contracts concerned with Web services are called Web services e-contracts (WS-contracts). The current complexity involved in e-contract establishment may hinder new business partnerships. The major related problems are [1], [2], [5]-[7]: the great amount of information necessary for e-contract establishment; the increasing number of parameters to be considered; the potential long-duration of electronic negotiations; and the involvement of different profiles of distinct organizations.

One way to overcome the mentioned e-contract drawbacks can be by tackling information structuring and reuse. Many e-contract establishment approaches use e-contract templates [2], [5]-[10] to overcome these issues. Templates, in most of them, are treated as simple documents that have empty fields to be fulfilled with some value, usually from a pre-defined list, during e-contract establishment. Moreover, the templates do not usually provide support for structured representation and selection of optional or alternative e-contract parts. In general, existing approaches do not offer suitable mechanisms to manage common and variable elements in similar e-contracts.

Product Line (PL) [11] promotes the generation of specific software products from a product family based on the reuse of a well-defined infrastructure. It exploits common points among systems in the same domain and manages variabilities among them in a systematic way. Commonly, feature modeling [12] is used to capture and manage such common points and variabilities [13]. Since WS-contract establishment demand an efficient information reuse, it can also take profit of the PL approach to achieve a better understanding of domains and reduce time-to-market.

This Ph.D. research proposes a new approach to reduce the complexity of WS-contract establishment. It is based on PL concepts, mainly on the feature modeling technique. Its major contribution is to offer a systematic and efficient way for information structure and reuse. The approach provides a mean to represent a WS-contract as a feature model that can be transformed into WS-contract templates. Moreover, it allows efficient management of mandatory, optional and alternative parts of WS-contracts. Two papers were published with intermediary results [14], [15].

In brief, the proposed approach consists of a set of five stages. Feature modeling allows the representation of generic e-services and possible levels for QoS attributes, which are mapped to a contract template. The WS-contract establishment process will be oriented by the feature model and their configurations. The generic e-services will be mapped to the Web services implementing them, in a one-to-one relationship.

2. Background concepts

The main concepts involved in this work are: electronic contracts and feature modeling. In this section, overviews of both concepts are presented.

2.1. Electronic contracts

A contract is an agreement between two or more parties interested in creating mutual relationships on business or legal obligations. It defines an activity set to be carried out by each party, which must satisfy a set of terms and conditions – known as contract clauses. An e-contract is an electronic document used to represent an agreement between organization partners carrying out business using the Internet, in which the negotiated services are e-services, currently implemented as Web services.

An e-contract consists of [2]: *parties* – representing the organizations involved in a business process; *activities* – representing e-services to be executed throughout the e-contract enactment; and *contractual clauses* – describing constraints to be satisfied throughout the e-contract enactment. Contractual clauses can represent three different types of constraints [1]: *obligations* – what parties should do; *permissions* – what parties are allowed to do; and *Prohibitions* – what parties should not do.

Obligations include QoS clauses associated with e-services which define attributes related to non-functional properties. They affect the definition and execution of an e-service, regarding to, for example: availability, integrity, reliability, performance, security and reply time [3], [4], [7]. For each QoS attribute, a value must be defined to be used as a tolerable level (e.g. a minimum, a maximum or an exact value).

2.2. Feature modeling

Feature modeling is an important technique that has been applied for capturing and managing commonalities and variabilities in software PL [11]. It was originally proposed in the domain engineering context, as part of the Feature-Oriented Domain Analysis (FODA) method [12], and has been applied in a range of domains including telecom systems, template libraries, networks protocols and embedded systems.

In general, a feature model is a description of the relevant characteristics of some entity of interest. A feature can be defined as a system property that is relevant to some stakeholder and is used to capture commonalities or discriminate systems in a family. They may denote any functional or non-functional characteristic at the requirement, architectural, component, platform, or any other level. According to the original FODA method, features can be mandatory, optional or alternative.

A feature model describes the configuration space of a system family. A member of the family can be specified by selecting the desired features from the feature model within the variability constraints defined by the model. This process is called feature configuration. The rules to elaborating features models or diagrams can be specified by feature metamodels.

Features can be organized in a feature diagram, which is a tree-like structure where each node represents a feature and each feature may be described by a set of sub-features represented as children nodes. Feature diagrams offer a simple and intuitive notation to represent variation points without delving into implementation details. The diagrams are especially useful to drive the feature configuration process.

3. Proposed approach

The proposed process, based on FORM process [13], consists of five stages, grouped in two life-cycle models, as follows:

I) Contract template development: carried out only once for a contract model:

1. <u>E-services feature model elaboration</u>: two feature models are elaborated to represent the e-services and QoS attributes from each one of both organizations willing to establish WS-contracts;
2. <u>WS-contract template creation</u>: having the two e-services feature models as the basis, a WS-contract template is created. It will contain basic information that can be used in any WS-contract to be established from there models;
3. <u>Web services development and publication</u>: Web services that implement the e-services to be electronically contracted must be developed and published to be available during WS-contract establishment;

II) Contract instance development: carried out once for each contract instance:

4. <u>E-services feature model configuration</u>: the two e-services feature models are then configured to represent the exact e-services and QoS levels for a particular business process between the two involved organizations;
5. <u>WS-contract establishment</u>: a WS-contract is produced by refining the WS-contract template, based on the previously defined pair of e-services feature model configurations.

Fig. 1 represents the artifacts produced throughout the stages above and the relationship between them. The `e-services feature model` is the basic artifact from which a unique `WS-contract template` is generated and one or more `e-`

`services feature model configurations` are derived. For each `e-services feature model configuration`, a particular `WS-contract` is established. The `WS-contracts` are established based on the same `WS-contract template`. Each `Web service` implementing an e-service of the feature model is referred to by the `WS-contract template`. Only the `Web services` implementing e-services of the feature model configuration are referred to by the corresponding `WS-contract`.

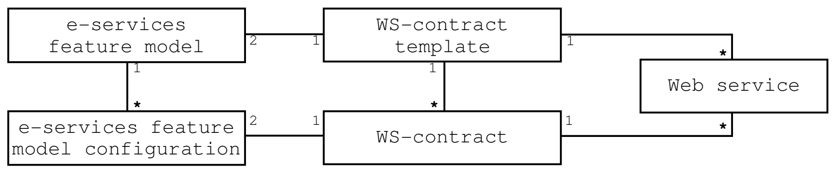

Fig. 1. Artifacts relationship

The e-services and QoS attributes are treated as common points and variabilities in feature models. They can be specified by mandatory, alternative and optional features. A feature metamodel [11] was chosen to drive the modeling of this information as features. And a specific feature diagram structure for e-services and QoS attributes representation is being developed, since the inherent flexibility of the metamodel would allow the definition of e-services and QoS attributes in too many ways.

A WS-contract metamodel was defined to represent rules to create both WS-contracts and templates. The metamodel was created by unifying the main concepts related to: (i) Web services – described by WSDL language; (ii) business processes involving Web services – described by WS-BPEL language; and, (iii) QoS of Web services – described by WS-Agreement language. No other similar type of metamodel was found in the literature, even regarding only WSDL and WS-BPEL.

The creation of the WS-contract template is carried out in two steps: at first, the WSDL and WS-Agreement sections are created directly from the e-services feature models. For the first step, there is a mapping from elements of the feature metamodel to elements of the WS-contract metamodel. In the second step, the WS-BPEL section is created from WSDL definitions and further information is defined during this stage.

To enable contract instantiation, the WS-contract template is instrumented with a set of annotations linking the contract elements to the respective features used as basis for its creation. During contract instantiation, the feature model configurations are used by a parser in a removal process. This process is driven by the mandatory features and the optional/alternative features that have been selected or not.

4. Supporting tool and case study

A support tool is being developed to aid the proposed process. The tool, named FeatureContract, includes a series of software components related to different stages of the approach. It is considered still a prototype tool and some challenges related to this development are: (i) identifying the main parts necessary to be finished, since its complete development is not feasible here; (ii) making possible to change the 3rd-party components, to be possible extend them; and (iii) defining the best strategy for component integration, since a set of several components is being used together.

A case study is being undertaken to evaluate the approach proposed in this Ph.D. thesis. It is concerned with the integration between two business and operation support systems, in the telecom context: customer relationship management (CRM) and dunning systems. The success on its execution (based on experimental software engineering concepts) is making possible to show the feasibility of the approach. The main challenge here is regarding the evaluation of its effectiveness, since comparing it to related approaches is not trivial.

Aiming at provide a rigorous validation of this proposed approach, the case study is being carried out based on the concepts of experimental software engineering and the evaluation of software engineering methods and tools [16]], [17]. As part of the methodology been applied, an evaluation plan was conceived, in which the following activities are detailed: selection of the experimental object, definition of the hypotheses, selection of the pilot project, identification of the comparison methods and schedule for the evaluation execution.

5. Related work

The majority of the projects in the e-contract area is related to the specification phase of business process and the post-specification phases (enactment and/or monitoring). Regarding the process specification, main attention is given to representing a business process already well-known to make it understood for both humans and machines.

In relation to e-contract establishment, there are several projects involved in this field. However, most of them use only metamodels as a basic and limited way to achieve information reuse. In some few cases, they also use templates as a more efficient way to achieve information reuse. Examples are [2], [5], [6], [8]-[10]. There are also some projects that work directly with QoS attributes, including [3], [4], [7].

Some works focus on the negotiation phase before specifying the business process, but they are commonly concerned with the process to be followed and the tools to be used during the negotiation between the parties. Some projects related to e-negotiation are presented in [18]-[20]. In these and other similar approaches, there is little emphasis in information reuse compared to the approach proposed by this work.

6. Conclusion

This Ph.D. research proposes a new approach to WS-contract establishment based on feature modeling. Its main contribution is allowing a better management of common and variable points found in similar WS-contracts; and information structure and reuse in a systematic way. Such improvement is achieved by the use of WS-contract templates associated with feature models representing e-services.

Although this work can be considered in an advanced phase, there are some challenges that must be treated so that it can be successfully finished. The main remaining one is to define how comparing the proposed approach with other related ones, since it differentiates essentially from them. Moreover, details about the support tool development must be defined, which depends on a deeply approach review.

7. References

1. Marjanovic, O., Milosevic, Z.: Towards Formal Modeling of e-Contracts. In: The 5th IEEE International EDOC Conference, pp. 59--68, IEEE Press, Seattle (2001)
2. Hoffner, Y., Field, S., Grefen, P., Ludwig, H.: Contract-Driven Creation and Operation of Virtual Enterprises. Computer Networks 37, 111--136 (2001)
3. Sahai, A., Machiraju, V., Sayal, M., Moorsel, A., Casati, F.: Automated SLA Monitoring for Web Services. In: Feridun, M., Kropf, P.G., Babin, G. (eds.) DSOM 2002. LNCS, vol. 2506, pp. 28--41. Springer, Montreal
4. Menasce, D.A.: QoS Issues in Web Services. IEEE Internet Computing 6(6), 72--75 (2002)
5. Krishna, P.R., Karlapalem, K., Dani, A.R.: From Contract to E-Contracts: Modeling and Enactment. Information Technology and Management 6(4), 363--387 (2005)
6. Hoffner, Y., Field, F.: Transforming Agreements into Contracts. International Journal of Cooperative Information Systems 14(2/3), 217--244 (2005)
7. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management 11(1), 57--81 (2003)
8. Chiu, D.K.W., Cheung, S.-C., S. Till.: A Three Layer Architecture for E-Contract Enforcement in an E-Service Environment. In: 36th HICSS, pp. 74, IEEE Press (2003)
9. Rouached, M., Perring O., Godart, C.: A Contract Layered Architecture for Regulating Cross-Organisational Business Processes. In: Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 410--415. Springer, Nancy
10. Berry, A., Milosevic, Z.: Extending Choreography with Business Contract Constraints. International Journal of Cooperative Information Systems 14(2/3), 131--179 (2005)
11. Czarnecki, K. et. al.: Staged Configuration through Specialization and Multi-Level Configuration of Feature Models. Software Proc.: Improv. and Prac. 10(2), 143--169 (2005)
12. Kang, K.C., Cohen, S., Hess, J., Novak, W., Peterson, A.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Tech. Report CMU/SEI-90-TR-021, SEI/CMU, (1990)
13. Kang, K.C. et. al.: FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures. Annals of Software Engineering 5, 143--168 (1998)
14. Fantinato, M., Toledo, M.B.F., Gimenes, I.M.S.: A Feature-based Approach to Electronic Contracts. In: IEEE 8th CEC and 3rd EEE, pp. 34--41, IEEE Press, San Francisco (2006)
15. Fantinato, M., Gimenes, I.M.S., Toledo, M.B.F.: Web Services E-contract Establishment Using Features. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 4102. LNCS, vol. 2506, pp. 290--305. Springer, Vienna
16. Kitchenham, B., Pickard, L., Pfleeger, S.L.: Case Studies for Method and Tool Evaluation. IEEE Software 12(4), 52--62 (1995)
17. Pfleeger, S.L.: Experimental Design and Analysis in Software Engineering: Part 2: How to Set up and Experiment. ACM SIGSOFT Soft. Eng. Notes 20(1), 22--26 (1995)
18. Streitberger, W.: Framework for the Negotiation of Electronic Contracts in E-Business on Demand. In: IEEE 7th CEC, pp. 370--373, IEEE Press, Munich (2005)
19. Rinderle, S., Benyoucef, M.: Towards the Automation of E-Negotiation Processes Based on Web Services - A Modeling Approach. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 443--453. Springer, New York
20. Kaminski, H., Perry, M.: SLA Automated Negotiation Manager for Computing Services. In: IEEE 8th CEC and 3rd EEE, pp. 47--54, IEEE Press, San Francisco (2006)

IT Service Management Processes for Concatenated Services in Multi-Domain Environments

Matthias Hamm

Munich Network Management Team,
Leibniz Supercomputing Centre (LRZ),
Boltzmannstr.1, 85748 Garching, Germany
matthias.hamm@lrz-muenchen.de

Abstract. The co-operation of several providers is still a challenge for IT Service Management. Currently, established IT Service Management approaches presume a hierachical organisational model. The growing complexity of novel IT services demands the development of alternative ways of provider collaboration. In our PhD thesis we focus on management problems raised by non-hierarchical co-operation in a multi-domain environment. For systematical analysis the concept of Concatenated Services is introduced. To foster the operations of these services, a specification method for service management processes is developed which incorporates approaches from organisational theory, together with a process model which guides the application of the method for concrete scenarios.

1 Introduction

Contemporary IT services, like triple-play broadband services for private homes or multi-national virtual private networks with bandwith up to several Gbps are way too complex and expensive to be provided by a single network provider. Instead, modern IT services are realized by a co-operation of several organisations, each provisioning a subset of the needed service functionality. Such *multi-domain environments* are still a central challenge for the field of IT Service Management (ITSM), which targets all activities to achieve and sustain a defined service quality for customers.

Hierarchical models of co-operation are quite well understood these days and are addressed by frameworks like the IT Infrastructure Library (ITIL) or the enhanced Telecom Operations Map (eTOM). Both of these frameworks assume that a single provider commissions one or more sub-providers to realise a service. In our work, we concentrate on service management processes in non-hierarchical environments, which are not well investigated yet. The concept of Concatenated Services is introduced to model a specific subset such services. To facilitate the co-operation of the providers, a specification method for the operational processes for Concatenated Services is developed, along with a process model which explains the application of the method for concrete scenarios.

The remainder of this paper is structured as follows: In the next section, the Géant2 End-To-End (E2E) Link service is introduced to discuss the specific management problems associated with Concatenated Services using a real-life scenario. In section 3 related work useful for our work is presented. Our approach is outlined in section 4. The paper ends with an outlook on the next steps.

2 Problem Statement

In most cases, a single provider takes over a prominent position in a multi-domain environment. This *hierarchical* organisational model has the advantage of clear contractual relations between the customers, the provider and its sub-providers. However, there are cases where the hierarchical model can not be applied, e.g. due to corporate-policies or funding issues. In this case several providers co-operate in delivering a service in a federated way without having a single provider taking over a steering position. This organisational model is called a *heterarchy* [1]. A representative example for an IT service provided within a heterarchical organisation is the Géant2 End-to-End (E2E) Link Service.

2.1 Géant2 End-to-End Links

The European Backbone Network Géant2 connects about 30 National Research and Educational Networks (NRENs) providing networking services to the scientific community [2]. Each NREN is an independent organisation which operates its network autonomously and has its own procurement policy. Thus, Géant2 is a representative example for a non-hierarchical multi-domain environment.

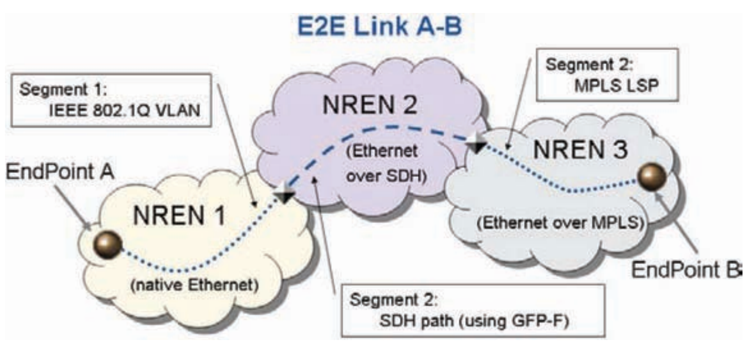

Fig. 1. Géant2 End-to-End Links

Besides IP services, with E2E Links Géant2 offers dedicated point-to-point connections at ISO/OSI Layer 2. These links are provided for international research projects like the Large Hadron Collider (LHC) at CERN. Figure 1 shows the typical structure of an E2E Link.

2.2 Concatenated Services

As a model for non-hierarchical services like Géant2 E2E Links, we define the class of Concatenated Services which have to satisfy the following characteristics:

Service Chaining A sequence of partial services at the same technical layer is needed to provide the whole service. The partial services can be realized using different technologies. They are provisioned and operated by several independent providers. In the scenario, each NREN contributes one or more segments of an E2E Link.

Heterarchy There is no hierarchy imposed on the relationship of the providers. In the scenario, due to the independence of the NRENs the collaboration of multiple administrative domains is needed to provide these links.

QoS Guarantees The service has to be provided with a specified quality, depending on the QoS parameters requested by the customers. For example, the LHC project has ambitious availability requirements for E2E Links. [3].

Only services satisfying *all* these characteristics are called Concatenated Services. For instance, Multi-domain IP connections in the Internet do not fit in this model, because they are provided on a best-effort basis only. E2E Links are a typical representative of the class of Concatenated Services, thus we will discuss the management problems of E2E Links in this broader context without confining our solution to the Géant2 scenario.

2.3 Management Processes of Concatenated Services

We assume that the providers form a long-term federation. In this organisational context, the specification of operational processes can foster the co-operation of the providers [4, 5]. The following specific issues have to be considered:

Roles and Responsibilities The allocation of roles is one of the most critical tasks in the co-operation of organisations. Which roles are necessary to perform tasks in federated operational processes and who takes over the responsibility for executing the individual tasks? How can functions like a service desk be realized in a heterarchic environment?

Interorganisational Processes Which processes are required for Service Management and how can they be structured? What are specific workflow patterns, like information exchange or coordination between the participants?

Binding Specifications In an multi-domain environment, binding regulations are necessary to enforce that all tasks are performed properly an to be able to conclude contracts. How can processes be specified so that the agreements between the providers are comprehensive for all participants on one hand and are clear without ambiguities on the other hand?

Control and Escalation The execution of the operational processes has to be observed to ensure that the processes are executed correctly and in a proper time frame. In the case of deviations or to be able to settle differences, escalation procedures have to be applied. How can the process control be structured in an heterarchical environment?

Process Automation Informal communication via e-Mail and phone is not well-suited to meet tight time restrictions in operational processes. How can interorganisational processes be supported by workflow management tools?

3 Related Work

In the last years, several approaches for Service Management have been propagated. ITIL [6, 7] and eTOM [8] have gained most appreciation and dissemination. Although these frameworks offer general concepts, they are focused on operational processes within a single provider organisation; interorganisational aspects are covered only for hierarchical provider co-operation. Non-hierarchical organisational models for the provisioning of IT services have barely been investigated, with the exception of individual projects like PREPARE, based on the the Telecommunications Management Network (TMN) framework [9].

In the broader context of economics, the co-operation of independent organisations gained increasing interest since quite a couple of years [10]; the characteristics of heterarchical co-operation has been analyzed e.g. by [1]. With the emergence of e-Business, the co-operation of companies based on information technology gained importance, covered by approaches like Business Process Management (BPM) [11]. As Service Management processes can be understood as the business processes of IT service providers, these approaches can be taken into account for our work, although they have to be adapted according to the requirements of Service Management.

The modeling and execution of business processes is covered by a couple of approaches, like Business Process Execution Language (BPEL) or the Business Process Modeling Notation (BPMN) [12, 13]. The Private-to-Public (P2P) approach to interorganisational workflows introduced by van der Aalst seems notably promising for our work, as it explictely supports the co-operation of independent organisations [14]. The e-Business consortium RosettaNET pursuits a similar approach, which is also taken into account for the further development of the eTOM framework [15, 16].

4 Modeling Processes for Concatenated Services

In the P2P approach, workflows within the individual organisations callled private workflows are separated from the overall public workflow. The details of the private processes are left to the partners while the public workflows are shared and commonly agreed between all organisations. Figure 2 exemplary shows a simplified Incident Management Process for Géant2 E2E Links. The overall process is defined as a public process. Each one of the numerous NRENs can implement its own private sub-processes, as long as they fit the requirements of the public process. However, the fully-fledged specification of the process using existing workflow languages leads to a very complex description, being hard to understand. For example, consider all the process details needed to specify the co-operation for first locating an error throughout the NRENs, then instructing

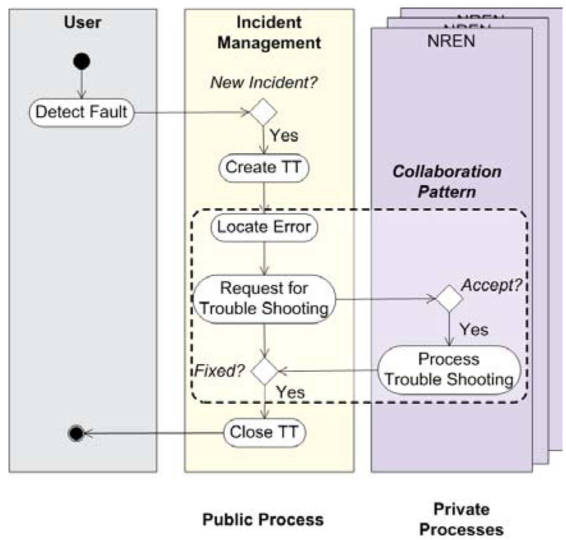

Fig. 2. Simplified Incident Management Process of Géant2 E2E Links

the affected NRENs to do the trouble shooting, and finally check that the actions have resolved the incident. In our work as a first step towards re-using established ITSM concepts we aim at developing a method which simplifies the specification of Service Management processes for Concatenated Services. Our work features two main contributions:

Co-operation Patterns To facilitate and simplify the specification of operational processes, co-operation patterns are defined covering the recurring parts of the processes. The patterns can be used as building blocks for the process descriptions. In Figure 2, patterns are needed for describing the collaborative trouble shooting.

Specification Process Model A Process Model provides instructions on how to specify operational processes for Concatenated Services. The Process Model will consist of the following three steps:

1. Scenario Analysis: In this step, the Service Management processes needed to meet the QoS requirements for a given scenario are determined.
2. Processes Specification: In the second step, the processes identified in the first step are specified in detail using Co-operation Patterns.
3. Process Automation: According to the required service quality the automation of the service processes can be necessary, e.g. to guarantee short processing times in case of service outages. For this purpose, the processes are implemented using worflow management tools.

5 Conclusions and Future Work

We have discussed in this paper the limitations of current solutions for service management which do not address non-hierarchical forms of provider cooperation. We introduced the concept of Concatenated Services and outlined our approach for the specification of operational processes for this class of service. We will elaborate this approach in our PhD Thesis. As a proof of concept, we plan to show the applicability for incident and problem management processes for Géant2 E2E Links.

References

1. Hedlund, G.: Assumptions of Hierarchy and Heterarchy, with Applications to the Management of the Multinational Corporation. In Ghoshal, S., Westney, E., eds.: Organizational Theory and the Multinational Corporation, London (1993)
2. Geant2: Homepage (2007) [Online: http://www.geant2.net/].
3. Large Hadron Collider Computing Grid (LCG): Homepage (2007) [Online: http://lcg.web.cern.ch/LCG/].
4. Kamminga, P.E., van der Meer-Kooistra, J.: Control patterns in joint ventures relationships. In Teillieu, T., ed.: Collaborative Strategies and Multi-organizational Partnerships. Garant, Leuven-Apeldoorn, The Nederlands (2001) 29 34
5. Hegering, H.G., Abeck, S., Neumair, B.: Integrated Management of Networked Systems Concepts, Architectures and their Operational Application. Morgan Kaufmann Publishers (1999)
6. Office of Government Commerce, ed.: Service Support. TSO, Norwich, GB (2000)
7. Office of Government Commerce, ed.: Service Delivery. TSO, Norwich, GB (2001)
8. Tele Management Forum: Enhanced Telecom Operations Map (eTOM). The Business Process Framework. Technical Report GB921 (2007)
9. Hall, J.: Management of Telecommunication Systems and Services, Modelling and Implementing TMN-Based Multi-Domain Management. Volume Vol. 1116 of Lecture Notes in Computer Science. Springer (1996)
10. Alter, C., Hage, J.: Organizations Working Together. Sage Library of Social Research 191. Sage, Newbury Park, London, New Delhi (1993)
11. van der Aalst, W.M., ter Hofstede, A.H., Weske, M.: Business Process Management: A Survey. In van der Aalst, W.M., ter Hofstede, A.H., Weske, M., eds.: Business Process Management: International Conference BPM 2003, Eindhoven. Volume 2678 of LNCS. (2003)
12. Object Modelling Group: Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification, dtc/06-02-01 (2006) [Online: http://www.omg.org/cgi-bin/apps/doc?dtc/06-02-01.pdf].
13. OASIS: Web Services Business Process Execution Language Version 2.0 (2007) [Online: http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf].
14. van der Aalst, W.M., Weske, M.: The P2P approach to Interorganizational Workflows. In: Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE 01). Lecture Notes in Computer Science, Vol. 2068. Springer, Berlin (2001) 140 156
15. RosettaNet: Homepage (2007) [Online: http://portal.rosettanet.org/cms/sites/RosettaNet/].
16. Tele Management Forum: Enhanced Telecom Operations Map (eTOM). The Business Process Framework. Addendum B. Technical Report GB921B (2007)

Quality-aware Self-Management for Service Processes in Service-Oriented Architectures

Steffen Bleul

Kassel University, Distributed Systems, bleul@vs.uni-kassel.de

Abstract. One of the salient features of Service-Oriented Architectures is that services can be deployed and removed at runtime. But service replacement and management for service processes is a demanding task in complex IT-Systems, especially under additional constraints like optimizing the Quality of Service of a service process. A self-managing system is desired but missing. We have already achieved self-healing and self-optimization with our service brokering system. We apply ontologies to discover service alternatives and their QoS. In this paper we not only present our work on quality-aware service discovery but also propose a self-manageable infrastructure for service processes. The infrastructure can be dynamically instantiated, configured and bound to management endpoints with semantic service discovery. The approach not only automates the binding of service management systems of multiple vendors but also the SLA monitoring and the ad-hoc integration of services in service processes.

1 Introduction

A Service-Oriented Architecture (SOA) is the ideal architecture for highly dynamic systems especially for complex enterprise IT-Systems. The business logic of an enterprise can be directly implemented as services and arranged to business processes in the form of service compositions. A SOA directly reflects the need for easy integration, reusability and change without affecting the course of business. When it comes to Service Management, any architecture for distributed applications forfeits its salient advantages without some kind of self-management. Service management must ensure a certain level of quality which requires fast recognition and handling of service problems throughout the SOA.

An important building block for a self-manageable system is the Semantic Web. We have developed ontologies to describe service processes, services and their QoS. They allow automatic quality-aware binding of services to service processes at runtime [1]. Whereas semantic service discovery is important for Service Management it is just the first step towards self-management. In this paper we present an approach for a self-manageable infrastructure. Newly deployed service processes, their management endpoints and management entities are automatically instantiated and connected. We apply semantic service discovery in order to support automatic configuration of the infrastructure. The required management model and components are introduced in Section 2. We present our achievements on quality-aware semantic service discovery for service processes in Section 3. In Section 4 we discuss related work and the paper closes with a conclusion in Section 5.

2 Self-Management for Service Processes

Our achievements (see Section 3) already enable self-properties inside a SOA and can be applied to self-management. The overall goal is a self-managing distributed infrastructure for quality-aware service process management by using semantic service discovery to plug management components of service consumers and service vendors together. Service processes must be deployable at runtime and services must be dynamically found and bound to the service process. Moreover, services must be managed with respect to their QoS which also requires dynamic instantiation and monitoring of Service Level Agreements (SLA). Management of these services must be handled in a platform- and organisational-independent way by providing runtime information to and accepting management actions from all participating organisations. Without doubt, the impact of this work is dependent on the ability to integrate the resulting system in current IT-Organisations. Therefore, the dynamic configuration of the infrastructure will have the ability to involve the local organisation's reporting, change management or manual administration.

The necessary distributed infrastructure for SLA and service monitoring and Service Management must even be able to be dynamically instantiated in order to support service replication and integration of newly outsourced or bought services. This requires a general model for Service Management. We describe Service Management as an interaction between management interfaces. The management interfaces deal with providing information about a service state, processing service information or to submit management actions to change a service's state. The interaction between management interfaces are described by its exchanging messages consisting of sensor information and management actions. Moreover, service process management is the accumulation of all management interfaces and interactions of the participating services and service managers. A service process can be managed in a platform- and organisational-independent way by routing sensory information to an arbitrary amount of management entities and by redirecting the resulting management actions to the Service Management interfaces. In our model we define three types of services to enable a self-managing infrastructure:

– **Sensor**: A sensor is a service that produces state information about a service or service process. A sensor produces its output independently of other sensors (low level sensors) or is in need of the input of other sensors (high level sensors). For instance a low level sensor sends the availability information of a single service to s high level sensor that evaluates the availability of the related service processes.

- **SLA Monitor**: An SLA Monitor is a specific high level sensor. The SLA Monitor accepts messages of sensors responsible for service quality dimensions, e.g. response time and price, and produces messages consisting of changes, warnings and violations. SLA Monitors are instantiated by SLA templates and can be connected hierarchical to ensure SLA-conformity throughout several service vendors and service processes.
- **Manager**: A Manager is a service that produces the transition from sensor information to management actions. We define a platform-independent vocabulary to describe service states, quality dimensions and management actions as a set of properties. The postconditions of the rules are mapped on management actions. A Manager can be instantiated by a management template and replicated for several services or service processes.

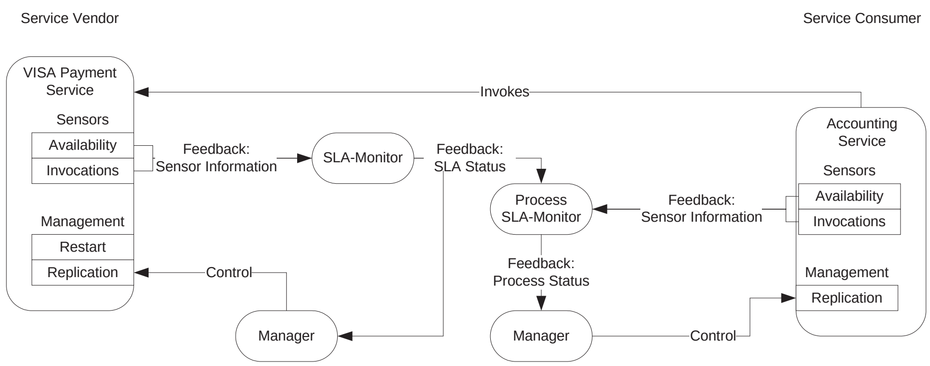

Fig. 1. Example Configuration of the Distributed Infrastructure.

We define self-management of service processes by a feedback and control loop where sensors forward their information messages through SLA Monitors to Managers and the Manager's management actions are sent back to service management interfaces. An example can be found in Figure 1. The infrastructure supports an arbitrary amount of Sensors, SLA Monitors and Managers. Apart from the development of standardized components of an SLA Monitor or Manager the remaining scientific challenge is the automatic configuration of the infrastructure.

3 Achievements

In the context of the DFG-funded project ADDO [2] we developed an ontology, a service discovery algorithm and system architecture for Semantic Service Discovery [1]. The resulting system not only discovers services but integrates them by solving mediation problems. The results have been proven to be important building blocks for automated service integration and have been published at several international conferences [1, 3, 4] and workshops [5, 6]. We have developed an ontology schema for describing service functional and non-functional properties. For the first time a semantic discovery system can be employed in a SOA. Moreover, the system's management interface can be used to realize self-properties, e.g. self-healing and self-optimization, inside a SOA which is discussed in [6]. We intend to apply the results in the following approaches:

- **Semantic Service Discovery**: The interfaces of services, sensors, SLA Monitors and Managers are described by semantic service descriptions. Semantic Service Discovery will be utilized to map sensor information on service processes and more specific to the participating services.
- **Semantic Service Composition**: Semantic Service Composition discovers an execution of auxiliary services that can transform sensor information, e.g. calculating performance for a service process. Here we apply our award winning semantic composition system [7].
- **Quality-aware Service Discovery**: Services are discovered with respect to their QoS. The matching result is used as a template to instantiate an SLA Monitor.
- **Semantic Service Mediation**: Semantic Service Mediation is applied to bridge syntactic differences between management systems, their exchanging messages and interface behavior.

The ADDO ontology can be divided into several sub-ontologies. The most important part is the *Service Level Ontology* introduced in [5]. Our ontology not only allows semantic description of a service's performance but also a Service Level Requirement (SLR) which describes the QoS requirement of service processes. Along with our discovery algorithm it allows quality-aware semantic service discovery. The algorithm automatically creates Service Level Agreements with respect to a service vendor's Service Level Offer and the service consumer's SLR. We can also reason about contract alternatives, like a cheaper version of a Service Level Agreement in exchange for an expense like lower performance.

The *Service Process Ontology* describes service processes along with service roles and their functionality [3]. The ontology allows reasoning for required services in conjunction to the service execution language BPEL4WS. Discovered services are bound to BPEL processes and the system can distinguish between several versions of the same description. Even if an existing BPEL process becomes unavailable due to faulting of participating services the system can switch to an older version or alternative process description.

Our extension of the OWL-S ontology and the implemented mediator address mediation capabilities between services [1]. Service interfaces may differ in syntactic details, e.g. different identifiers for parameters, different service operations. Furthermore, our ontology schema offers the flexibility to specify aspects like optional but not necessary service operations and parameters. In most cases such differences can be bridged with a service mediator at runtime. These mediators are realised as proxy services and developed in an EAI project. Their cumbersome development causes a major lack of flexibility inside a SOA. The implementation even takes different message formats into consideration. This technique allows automatic message transformation which can solve most mediation problems that occur with Web Services.

4 Related Work

Web Service standards are especially important for this work. The OASIS WSDM (WSDM) [8] consists of specifications for Web Service Endpoints and standardized messages for Service Management. We use them to define service interfaces and messages formats for the resulting infrastructure. Also the OASIS Web Service Notification (WSN) [9] standard is an important part as it offers a model to disseminate data, events and management actions.

We use IBM's Web Service Level Agreements (WSLA) [10] and the Grid Communitys Web Service Agreements (WS-A) [11] to develop our SLA Monitor. They are related to our SL-Ontology. They define a schema for the XML-based description of SLAs. Furthermore, WS-A defines interfaces and message formats for SLA-Management and a Web Service implementation named Cremona [12].

The projects METEOR-S [13] and AMIGO also develop ontologies and apply semantics for Service Management. Their results are especially important for quality-aware service discovery and for developing ontologies for service processes but they do not define the required platform- and organisational-independent distributed management infrastructure.

The Dialogue Interaction Specification Language (DISL) [14] is the base for our service manager. It includes a control model consisting of rules, data types and flow control to specify behaviour and an interpreter implemented as a service. We are currently adjusting DISL for Service Management in order to specify management behaviour without detailed knowledge of the exchanged messages and management endpoints. In [15] we have already shown an example on how the interpreter can be used for interaction driven control of distributed components.

5 Conclusion

In order to enable self-manageable service processes we propose a standardized infrastructure consisting of management interfaces, message formats and management components and to apply semantic service discovery to bind them to their respective management endpoints. This allows not only immediate deployment of service processes and enforcement of their contracts but also the utilization of management interfaces of multiplies parties. The development of the required building blocks on existing standards and demands in IT-Organisations is a challeging task but also comply with our achievements and proceedings.

References

1. Bleul, S., Zapf, M., Geihs, K.: Flexible Automatic Service Brokering for SOAs. In: Proceedings on 10 th IFIP / IEEE Symposium on Integrated Management (IM 2007), Munich, Germany (2007)
2. Geihs, K., Bleul, S.: (Automatic Service Brokering in Service oriented Architectures) Project Homepage: http://www.vs.uni-kassel.de/research/addo/.
3. Bleul, S., Zapf, M., Geihs, K.: Automatic Service Process Administration by Semantic Service Discovery. In: Proceedings of the 7 th Conference Internationale sur les Nouvelles Technologies de la Repartition (NOTERE 2007), Marrakech,Maroc (2007)
4. Bleul, S., Weise, T., Geihs, K.: Making a Fast Semantic Service Composition System Faster. In: Proceedings of the 2007 IEEE Joint Conference on E-Commerce Technology (CEC'07) and Enterprise Computing, E-Commerce and E-Services (EEE'07), Tokyo, Japan (2007)
5. Bleul, S., Weise, T.: An Ontology for Quality-Aware Service Discovery. In: Proc. First International Workshop on Engineering Service Compositions (WESC'05). (2005) 35–42
6. Bleul, S., Zapf, M.: Ontology-Based Self-Organization in Service-Oriented Architectures. In: Proceedings of the Workshop SAKS 2007, VDE (2007)
7. Bleul, S., Weise, T., Geihs, K.: Large-scale service composition in semantic service discovery. In: Proceedings of the IEEE Joint Conference on E-Commerce Technology (CEC'06) and Enterprise Computing E-Commerce and E-Services (EEE'06), Springer Press (2006) 427–429
8. Murray, B., Wilson, K., Ellison, M.: Web Services Distributed Management (WSDM). Organization for the Advancement of Structured Information Standards (OASIS). (2006)
9. Graham, S., Hull, D., Murray, B.: OASIS Web Services Notification (WSN). Organization for the Advancement of Structured Information Standards (OASIS). (2006)
10. Dan, A., et al.: Web Service Level Agreement (WSLA) Language Specification. International Business Machines Corporation (IBM). (2002)
11. Aiello, M., Frankova, G., Malfatti, D.: What's in an Agreement? An Analysis and an Extension of WS-Agreement. Int. Conf. on Service-Oriented Computing (ICSOC) (2005)
12. Ludwig, H., Dan, A., Kearney, R.: Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreents. In: Proceedings of the International Conference on Service Oriented Computing (ICSOC'04), ACM (2004) 65–74
13. Verma, K., et al.: The METEOR-S Approach for Configuring and Executing Dynamic Web Processes, LSDIS Technical Report (2005)
14. Schaefer, R., Mueller, W., Bleul, S.: A Dialog Model for Multi Device Interfaces with Different Modalities. In: Proceedings of Human Computer Interaction (HCI) International 2005. (2005)
15. Schaefer, R., Bleul, S.: Towards Object Oriented, UIML-based Interface Descriptions for Mobile Devices. In: Proceedings of the 6th International Conference on Computer-Aided Design of User Interfaces (CADUI2006). (2006)