Florian Daniel
Fethi Rabhi
(Eds.)

ICSOC/ServiceWave 2009 PhD Symposium

Co-located with the
7th International Joint Conference on Service Oriented Computing

Stockholm, Sweden, November 26, 2009

Proceedings

Sponsor: IBM Research, USA

Preface

The Web has affirmed its undisputable role as communication platform not only for human actors, for instance, exchanging emails or instant messages, but also for software actors interacting via protocols like HTTP, SOAP, SMTP, or the like. In this context, today's industrial and academic research efforts are characterized by a huge interest in service-oriented technologies, models, architectures, paradigms, etc., especially for the development of distributed, web-based services and applications.

In its beginning, the service-oriented computing community mainly focused its attention to the problem of abstracting from implementation details and devising technology-agnostic communication protocols, such as SOAP (and its various extensions), in order to enable reliable communication and exchange of information among software actors. Then, the attention was (and still is) captured by the problem of service composition, yielding a variety of approaches (e.g., declarative, semantic, goal-driven, adaptive, self-healing) and a de-facto standard like BPEL. Nowadays, the novel challenges are called lightweight or user-oriented composition, hybrid composition of services and user interfaces ala web mashups, software as a service, cloud computing, and so on. Besides these technical challenges, there are many additional research issues related to the adoption, management and governance of service-oriented systems.

The ICSOC/ServiceWave PhD Symposium 2009 aims to bring together excellent PhD students working on topics that are related to service-oriented computing and to discuss the ideas, solutions and trends of tomorrow. Specifically, the Symposium is an international forum for PhD students working in all the areas addressed by the ICSOC/ServiceWave 2009 conference (http://www.icsoc.org/). The Symposium operates in a workshop format and gives PhD students the opportunity to showcase their research and particularly encourages students that are still developing their research methodology or are somewhere in the middle of their research program to submit their contributions for discussion.

The goals of the ICSOC/ServiceWave PhD Symposium are:

- To bring together PhD students and established researchers in the field of service oriented computing;
- To enable PhD students to interact with other PhD students and to stimulate an exchange of ideas, suggestions, and experiences among participants;
- To give PhD students the opportunity to present and discuss their research in a constructive and critical atmosphere;
- To provide students with fruitful feedback and advice on their research approach and thesis.

The ICSOC/ServiceWave PhD Symposium 2009 in Stockholm is the 5th PhD Symposium of a series held in conjunction with the ICSOC conferences in Sydney, Australia (2008), Vienna, Austria (2007), Chicago, USA (2006), and Amsterdam, the Netherlands (2005).

This year's edition attracted 22 submissions, out of which 10 high-quality contributions have been accepted for presentation during the Symposium. Contributions cover almost all aspects of the software lifecycle, spanning from design time issues, such as model checking for service compositions, security, guided and automatic service composition, to testing, maintenance, and evolution of service-oriented applications. Accepted papers have been submitted from 7 different countries (Austria, China, Denmark, Germany, Italy, The Netherlands, and the United Kingdom), affirming the growing role of China in the context of service-oriented computing, next to the traditionally Europe-centered audience of the ICSOC/ServiceWave conference.

Acknowledgments. First of all, we would like to thank all the students who submitted their manuscript for evaluation to the PhD Symposium; unfortunately, we were not able to accept them all. Then, we need to thank the organizers of the ICSOC/ServiceWave 2009 conference for their assistance with the preparation of the Symposium and the proceedings, and all the PC members for their timely and professional reviews and their participation in the discussion with the students.

Sponsorship. Finally, also this year's PhD Symposium could count on the sponsorship by IBM T.J. Watson Research Center, New York, USA. Thanks a lot for the support!

Florian Daniel
Fethi Rabhi

Trento, Sydney
November 26, 2009

Organization

PhD Symposium Chairs

Florian Daniel, University of Trento, Italy
Fethi Rabhi, UNSW, Australia

Programme Committee Members

Karim Baina, ENSIAS, Morocco
Claudio Bartolini, HP Labs, Palo Alto, USA
Salima Benbernou, University of Lyon, France
Athman Bouguettaya, CSIRO, Australia
Christoph Bussler, BEA, USA
Fabio Casati, University of Trento, Italy
Joseph Davis, University of Sydney, Australia
Marlon Dumas, University of Tartu, Estonia
Schahram Dustdar, Technical University of Vienna, Austria
Willem-Jan van den Heuvel, Tilburg University, The Netherlands
Dimka Karastoyanova, University of Stuttgart, Germany
Nikolay Mehandjiev, University of Manchester, UK
Hamid Motahari, HP, USA
Klaus Pohl, University of Duisburg/Essen, Germany
Omer Rana, University of Cardiff, UK
Regis Saint-Paul, CREATE-NET International Research Center, Italy
Farouk Toumani, University Blaise Pascal, France
Yannis Velegrakis, University of Trento, Italy

Table of Contents

Verification of Services Specified in BPEL

Joseph C. Okika*

Supervised by Prof. Anders P. Ravn
Computer Science Dept. Aalborg University, Aalborg, Denmark

Abstract. This project investigates, implements and evaluates tool support for analysis of SOA-Based service contracts using Model Checking. The specification language for the contract is Business Process Execution Language (BPEL). It captures the behavior of services and allows developers to compose services without dependence on any particular implementation technology. A behavior specification is extracted from a BPEL program for formal analysis. One of the key conditions is that it reflects the intended semantics for BPEL, and in order to make it comprehensible, it is specified in a functional language. The resulting tool suite is hosted on an Eclipse platform.

1 Research Question and Its Significance

Service Oriented Architectures (SOAs) are applicable when multiple applications running on varied technologies and platforms need to communicate with each other. In this way, enterprises can mix and match services to perform business transactions with less programming effort. However, a service operates under a contract/agreement which will set expectations, and a particular ontological standpoint that influences its semantics [14]. Services are first class citizens and are autonomous as well as distributed in nature. They can be composed to form higher level services or applications to solve business goals. Of course, this raises a lot of issues such as managing composed services, monitoring their interaction, analyzing the behavior of interacting services, verifying the functionality of individual services as well as composed services.

One of the most widely used languages for specifying services is BPEL [2]. It offers a programming model for specifying the orchestration of web services through several activities. Activities are categorized into two; basic and structured. Basic activities (for instance invoke, receive, etc.) define the interaction capabilities of BPEL processes whereas the structured activities are made up of constructs such as flow (for synchronization), compensate, and pick among other activities all of which can be inside a scope.

The interaction of all the features makes the behaviour of BPEL programs hard to understand. Consider a scenario in which a scope named A contains two scopes named B and C, which both have compensation handlers. Assume that

* Partially supported by the Nordunet3 project COSoDIS - “Contract-Oriented Software Development for Internet Services“. E-mail: ojc@cs.aau.dk

C has completed successfully and therefore can be compensated. However, an activity in scope B throws a fault that propagates to a fault handler in A:

$$\langle A \ldots \langle C \ldots C_{comp} \rangle ; \langle B \ldots Fault \ldots \rangle \ldots A_{Faulthandler} \rangle$$

If the A fault handler invokes compensation to undo the effect of the inner blocks, it is not done, because B has not terminated successfully. Its compensation handler is not active, thus its predecessor C cannot be compensated.

Traditional testing would hardly catch such a scenario, because the fault might be thrown at runtime by a dynamically composed service. Thus, there is a clear need to employ and integrate successful analysis techniques like model checking in the design of support tools for high quality SOA-based services.

Current Results In [5] we have demonstrated a viable solution to the problem of checking for some functional and behavioural properties of individual services. This is done through translation of the specifications to timed automata followed by model checking for relevant properties. In [3] we consider the problem of consistency across specifications and identified a need to set up a correspondence between the individual automata. The novel contribution in that paper is to make such a consistency check practical by translating the automata to CCS, the input language for the Concurrency Work Bench. As demonstrated by a case study, this technique is applicable and gives a handle for automating yet another consistency check for web services.

In [12] we give a classification of service contract specification languages based on application families and aspects. The classification identifies competing languages across aspects. It shows where a language may fit into the development of service based applications as well as the ones that allow for desired analyses, for instance match of functionality, protocol compatibility or performance match. In addition, we use the classification to survey analysis approaches. Furthermore, the classification may assist in planning of development activities, where an application involves services with contracts that span across families. Such scenarios are to be expected as service oriented applications spread. Another paper [4] focuses on analyzing behavioral properties for web service contracts formulated in Business Process Execution Language (BPEL) and Choreography Description Language (CDL). The key result reported is an automated technique to check consistency between protocol aspects of the contracts. The contracts are abstracted to (timed) automata and from there a simulation is set up, which is checked using automated tools for analyzing simulation relations.

2 Current knowledge and the existing solutions

In this section, we present several efforts geared toward formalizing/analyzing services specified using BPEL. The overall observation about these works is that they all deal with three major issues; semantics definition, mapping to an analysis language and applicability. In Figure 1, δ represents those efforts that

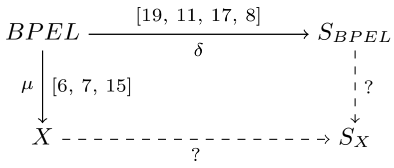


Fig. 1. BPEL Formalizations

cover semantics definition and mostly applying Petri net simulation while μ represents those that focus on mapping/translation to an analysis language.

Although most of these results cover only fragments of the language and some say little about the underlying analysis language and automation, tehy pave a way for a potential exploration of simulation in the case of Petri net based models and several kinds of analyses for automata based models.

The work with Petri net seems most complete and is an important yardstick. However, Petri nets give very very general models and it may be difficult to find subsets that allows model checking for desirable properties. Thus we shall try another road with a semantic preserving mapping of full BPEL activities to the UppAal tool leveraging the underlying transition system semantics of UppAal as well as the automatic verification engine. This will by necessity be an abstraction or an over-approximation of the functional parts and for some unbounded dynamic activities.

Work closer related to ours include: Abstract state machines which are used in [16] to define an abstract operational semantics for BPEL for version 1.1. The work focuses on formal verification of service implementations and resolving inconsistencies in the standard. Abouzaid and Mullins [1] propose a BPEL-based semantics for a new specification language based on the π-calculus, which will serve as a reverse mapping to the π-calculus based semantics introduced by Lucchi and Mazzara [10]. Their mapping is implemented in a tool integrating the toolkit HAL and generating BPEL code from a specification given in the BP-calculus. Unlike in our approach, this work covers the verification of BPEL specifications through the mappings while the consistency of the new language and the generated BPEL code is yet to be considered. As a future work, the authors plan to investigate a two way mapping. We look forward to seeing this.

Several model checking approaches have been employed to provide some form of analysis. An overview of most of the semantics foundation is given in [18]. An illustrative example which is well-explained is [11]. It deals with specification of both the abstract model and executable model of BPEL. The approach is based on Petri nets where a communication graph is generated representing a process's external visible behavior. It verifies the simulation between concrete and abstract behavior by comparing the corresponding communication graphs. Continuing with Petri net, an algebraic high-level Petri net semantics of BPEL is presented in [17]. The idea here is to use the Petri patterns of BPEL activities in model checking certain properties of BPEL process descriptions. The model is feature complete for BPEL 1.1. Lohmann extends this work with a feature-complete Petri net semantics for BPEL 2.0 [8]. As there exists several BPEL formalizations including a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures presented in [19, 13] a detailed comparison and evaluation of Petri Net semantics for BPEL is presented in [9].

In the case of using labeled transition systems as models for formalizing BPEL, a few efforts is found in the literature. They focus on some fragments of BPEL constructs. For instance, Geguang et al. present a language μ-BPEL [15] where a full operational semantics using a labeled transition system is defined for this language and it is mapped to Extended Timed Automata. Fu et al. presented a translation from BPEL to guarded automata in their work [6]; the guarded automata is further translated into Promela, which is the language for the SPIN model. Similar approaches are also followed in [7] but the mapping is not semantic preserving. All these efforts points to the fact that there is an important need for service contracts to be specified and analyzed.

3 Proposed Ideas

As mentioned in the previous section, many researchers have considered semantics of BPEL and its use for analysis. However, there are several issues around these semantics. First, there is the issue of coverage - that is to say, is the full BPEL language covered or some fragments of it? Most of the efforts using automata as presented in the previous section covers only some fragments of BPEL. A few of the efforts using Petri net covers a feature-complete BPEL. Our framework is an alternative based on automata (LTS). Second, there is the issue of translation where one may ask: is it semantic preserving? There is also the third issue of whether it is manual, semi-automated or automated.

Figure 2 shows the proposed approach; mapping BPEL to timed automata (TA). This defines a semantics for BPEL with a clear description of what is included and what is abstracted in the mapping and thus answers the issues raised above.

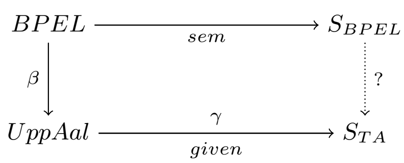


Fig. 2. The new Approach

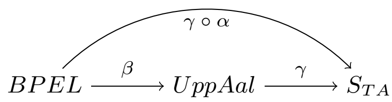


Fig. 3. Functional Composition

Looking at Figure 2, starting from BPEL, we consider a full behavior of BPEL syntax and define the semantics based on UppAal, S_{BPEL}. We follow a functional approach where we define a function β mapping BPEL to UppAal. We use timed automata for the formal model but with a rendering to UppAal because it is a mature model checking tool with wider audience and supported in our research environment. It can be a different choice (for example SPIN) in another environment. Note that the function *given* which takes care of the TA semantics is given with the UppAal tool and it's transition system semantics. Composing these two functions as shown in Figure 3 relates $BPEL$ to S_{TA}. In effect, having defined the function mapping BPEL to UppAal, we achieve a semantic preserving extraction/translation. That is, taking the inverse of the function gives us the result.

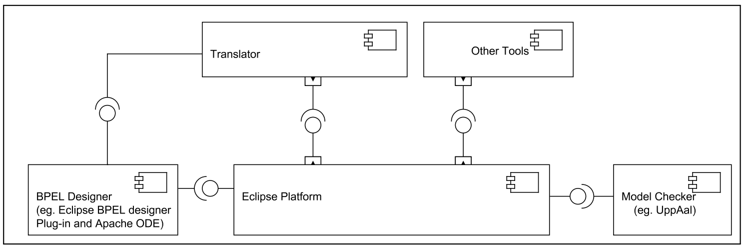


Fig. 4. Plug-in Architecture

4 Contribution to the problem domain and Discussion

The project offers three distinct contributions in the development of analysis and verification tools for SOA-based services. 1) The technique employed is a rigorous use of the power of functional languages in defining a property preserving mapping for the full behavior of BPEL. 2) Model checking of behavior properties of BPEL. General properties such as those related to deadlock and reachability as well as application specific properties are considered. Eg., services should not deadlock even with faults and compensation. 3) A prototype Integrated tool. The supporting tool will allow developers to leverage the already existing IDE such as Eclipse to design, specify and analyze SOA-based services.

Tool Development: We focus on building a theory based tool that gives developers of SOA-based services a clear understanding of BPEL processes. We consider SML, a functional language to be very suitable for transformations. An alternative is to use the XSLT transformation proposed by OMG but its verbosity and XML syntax make it hard to understand and one relies also on the transformation engines which again is hard to understand what is going on. We are implementing the integrated supporting tool as a plug-in in the Eclipse framework. A model (UML) of the various components of the analysis tool is shown in Figure 4.

Discussion: The main novelty is to solve the issue of semantics unrelated to analysis tools. This is achieved by defining the extraction function using a functional language. As a side effect to this, we develop a functional XML parser/unparser for Standard ML. As this is an ongoing work, further effort will be geared toward tuning the tool. We plan to build a service based point of sale system using ActiveVOS orchestration system to demonstrate analysis of properties.

References

1. F. Abouzaid and J. Mullins. A Calculus for Generation, Verification and Refinement of BPEL Specifications. *ENTCS*, 200(3):43–65, 2008.
2. A. Alves, A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, Sterling, D. König, V. Mehta, S. Thatte, D. van der Rijn, P. Yendluri, and A. Yiu. Web Services Business Process Execution Language Version 2.0. OASIS Std., 2007.
3. E. Cambronero, J. C. Okika, and A. P. Ravn. Analyzing Web Service Contracts - An Aspect Oriented Approach. In *Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'2007)*, pages 149 – 154. IEEE Computer Society Press, November 2007.
4. E. Cambronero, J. C. Okika, and A. P. Ravn. Consistency Checking of Web Service Contracts. *Int'l Journal Advances in Systems and Measurements*, 1(1):29–39, 2008.
5. G. Diaz, J. J. Pardo, M. E. Cambronero, V. Valero, and F. Cuartero. Verification of Web Services with Timed Automata. In *Proceedings of First International Workshop on Automated Specification and Verification of Web Sites*, volume 157, pages 19–34. Springer Verlags Electronics Notes Theor. Computer Sci. series, 2005.
6. X. Fu, T. Bultan, and J. Su. Analysis of interacting BPEL web services. In *Proc. 13th Int'l Conf. WWW'04*, pages 621–630, NY, USA, 2004. ACM.
7. X. Fu, T. Bultan, and J. Su. WSAT: A Tool for Formal Analysis of Web Services. In *Proc. 16th Int. Conf. Comp. Aided Verification*, pages 510–514. Springer, 2004.
8. N. Lohmann. A feature-complete Petri net semantics for WS-BPEL 2.0. In K. v. Hee, W. Reisig, and K. Wolf, editors, *Proc. Workshop on Formal Approaches to Business Processes and Web Services*, pages 21–35. University of Podlasie, June 2007.
9. N. Lohmann, H. Verbeek, C. Ouyang, and C. Stahl. Comparing and evaluating Petri net semantics for BPEL. *IJBPIM*, 2008. (Accepted for publication).
10. R. Lucchi and M. Mazzara. A pi-calculus based semantics for WS-BPEL. *Journal Log. Algebr. Program.*, 70(1):96–118, 2007.
11. A. Martens. Consistency between executable and abstract processes. In *EEE '05: Proceedings of the 2005 IEEE International Conference on e- Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service*, pages 60–67, Washington, DC, USA, 2005. IEEE Computer Society.
12. J. C. Okika and A. P. Ravn. Classification of SOA Contract Specification Languages. *Web Services, IEEE International Conference on*, 0:433–440, 2008.
13. C. Ouyang, E. Verbeek, W. M. P. van der Aalst, S. Breutel, M. Dumas, and A. H. M. ter Hofstede. Formal semantics and analysis of control flow in WS-BPEL. *Sci. Comput. Program.*, 67(2-3):162–198, 2007.
14. R. Perrey and M. Lycett. Service-oriented architecture. *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, pages 116–119, 2003.
15. G. Pu, X. Zhao, S. Wang, and Z. Qiu. Towards the Semantics and Verification of BPEL4WS. *Electr. Notes Theor. Comput. Sci.*, 151(2):33–52, 2006.
16. D. F. W. Reisig. ASM-based semantics for BPEL: The negative Control Flow. In *Proc. 12th Int'l Workshop on Abstract State Machines*, pages 131–151, 2005.
17. C. Stahl. A Petri Net Semantics for BPEL. Informatik-Berichte 188, Humboldt-Universitt zu Berlin, July 2005.
18. F. van Breugel and M. Koshika. Models and Verification of BPEL, 2005. http://www.cse.yorku.ca/ franck/research/drafts/tutorial.pdf.
19. M. T. Wynn, H. M. W. Verbeek, Aalst, T. A. H. M. Hofstede, and D. Edmond. Business process verification - finally a reality! *Business Process Mgmt. Journal*, 15(1):74–92, 2009.

Cross-organizational Service Security – Solutions for Attack Modeling and Defense

André Miede
Supervised by Ralf Steinmetz

Multimedia Communications Lab (KOM) – Technische Universität Darmstadt
Department of Electrical Engineering & Information Technology
Merckstraße 25, D-64283 Darmstadt, Germany
Andre.Miede@KOM.tu-darmstadt.de

Abstract Security is an important aspect of Service-oriented Architectures (SOAs), enabling the service-based integration of partner IT systems across organizational boundaries, i. e., in the Internet of Services. Current trends in SOA security, e. g., reducing it to Web service security, do not take into account SOA-specific threats, vulnerabilities, and attacks. In this paper, measures to support the modeling of attacks in general and in order to show the service-oriented difference regarding security are introduced. Based on this understanding, mechanisms to defend against SOA-specific attacks will be designed and evaluated.

1 Introduction

Challenging market dynamics and the rise of complex value networks require organizations to adjust their business processes rapidly in order to stay competitive. As many organizational processes are supported or even enabled by information technology (IT), a process is only as flexible as its underlying technological representation. A special integration challenge in this context are processes which span across organizational boundaries, e. g., customer creation processes, where data has to be checked against external watch lists in order to fight organized crime. Another example are trading processes in investment banking, where market data or credit ratings are bought from external providers.

The *Service-oriented Architecture* paradigm (SOA) [1] offers possibilities on both a technological and organizational level to integrate company-wide and inter-company IT systems. SOAs are based on the "service" concept, where services can be seen as black boxes representing business functionalities. These services are used to assemble business processes as service compositions and may even cross enterprise boundaries, thus, being *cross-organizational service-based workflows* [2, 3], e. g., in the *Internet of Services* scenario [4].

The rest of the paper is structured as follows: Section 2 presents the problem statement and the research questions which are at the foundation of my thesis. Section 3 structures both preliminary results and open challenges for the proposed questions. Section 4 concludes the paper and gives an outlook on next steps.

2 Problem Statement and Research Objectives

Just as any economic system requires security in order to work and to be accepted by its participants, the security of the involved systems, exchanged messages, and used communication channels has to be ensured for cross-organizational service-based collaboration. Achieving and guaranteeing basic IT security goals such as confidentiality, authentication, authorization, non-repudiation, integrity, and availability [5–7] is an absolute must in this context and still an active topic, both in research and industry. Thus, the differences SOA introduces into the field of IT security have to be analyzed and addressed.

The main tenor of current SOA security research is that conventional security measures are not sufficient in the SOA context [2, 8–10]. For example, a major argument in this context is the necessity to switch from point-to-point-security to end-to-end-security, because any used service can call an arbitrary number of different services on its own. Another argument is the need for decoupled security decision points, in SOA usually called security-as-a-service. Yet another trend is to equalize SOA security with Web service security, reducing SOA security requirements to Web service security standards and their configuration.

While these approaches are important building blocks for SOA security, they are not sufficient as they do not take into account *SOA-specific* threats, vulnerabilities, and the corresponding attacks. In order to close this gap, the following research challenges and objectives were identified:

1. Analyze SOAs regarding security challenges and specific attack scenarios, e. g., for the Internet of Services. The analysis must not be limited to particular SOA implementations, i. e., Web services, but focuses on SOA characteristics such as loose coupling, composability, etc. Based on the analysis of these security impacts, SOA-specific attacks have to be identified and modeled.
2. Develop means to understand and model attacks in general, i. e., analyze and define the elements they consist of. This objective is not service-specific, but a general IT security challenge. The results of this objective are used to model the SOA-specific attacks identified in the first objective.
3. Provide technology-independent solutions to defend against SOA-specific attacks. Based on the modeled attack scenarios, selected countermeasures have to be developed or adapted from other areas of research in order to make cross-organizational SOA scenarios safer.

The next section discusses my proposed solutions and their expected outcomes for theses challenges.

3 Proposed Solutions and Expected Outcomes

Addressing the research challenges and objectives outlined in Section 2, my research focuses on the following solution building blocks as depicted in Figure 1. In the following, for each of these solution building blocks, first results, their impact, and the progress beyond the state of the art is briefly discussed.

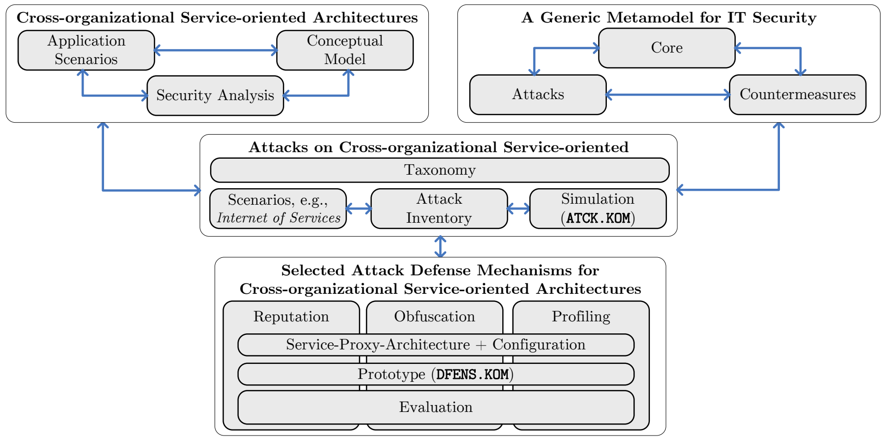


Figure 1. Research structure and approach

3.1 Cross-organizational Security – The Service-oriented Difference

Cross-organizational SOA security deals with the application of core IT security concepts such as threats, vulnerabilities etc. on the elements of cross-organizational SOA such as loose coupling, composability, etc. These elements are assembled in the form of a conceptual model which is based on SOA definitions and descriptions found in standard literature on SOA [2, 3, 11–13] (cf. Figure 2). The goal is to evaluate the security impact of single SOA elements and their relationships. While single security aspects of these elements are already well-known, i. e., for distributed systems characteristics, the combination of and relationships between the SOA elements as well as their impact makes cross-organizational SOA a special security challenge [14]. An example for such an impact is compromising an organization's legal or regulatory *compliance*, which can result in fines, the revocation of licenses, or loss of customer trust. *Composability* has also a high security impact, creating seams for exploitation, e. g., caused by the incomplete integration of different security technology, or by the possibility to introduce malicious services into the composition.

Compared to standard literature on SOA [2, 3, 11–13], which differ in their definitions, presentation, and coverage of SOA core elements, this approach offers a compact and visual inventory as a basis for communication and analysis, i. e., a dedicated SOA security analysis.

3.2 A Generic Metamodel for IT Security

The proposed generic metamodel for IT security [14] brings together the most important ideas of IT security (including their relationships) and consists of three main parts: a *Core* of basic IT security concepts, *Attacks*, and *Countermeasures*.[1] It lays the foundation for describing and understanding the different elements of attacks and countermeasures in an IT security context.

[1] http://www.kom.tu-darmstadt.de/~miede/soasecurity/secmetamodel.pdf

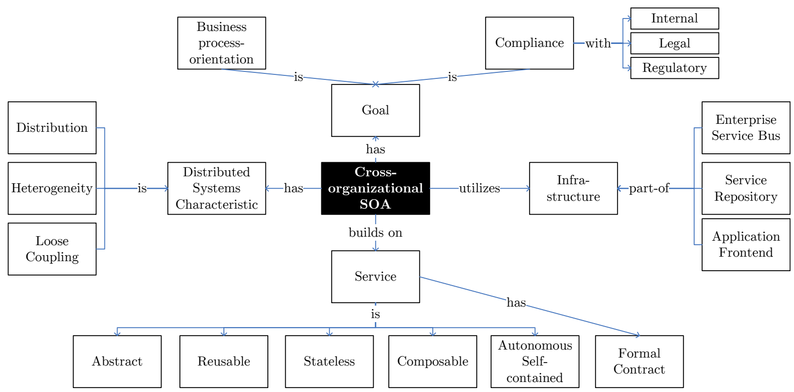


Figure 2. Conceptual model of cross-organizational SOA [14]

In order to show the applicability of the metamodel to real-life scenarios, attacks on different distributed systems were modeled, i. e., on Peer-to-Peer systems, on Mobile ad hoc Networks, and in SOA contexts. Compared to similar concepts such as attack patterns [15,16] or security patterns [17], this metamodel offers building blocks which help with actually assembling such patterns, thus, improving the means to understand and model attack knowledge.

3.3 Attacks on Cross-organizational Service-oriented Architectures

Cross-organizational SOA-specific attacks target vulnerabilities of single SOA elements and combinations thereof (cf. Figure 2). All types of attacks which are already known from classic distributed systems or which focus on specific technologies, i. e., Web services, are not considered cross-organizational SOA-specific attacks. This is due to the fact that these attacks can also occur outside SOA contexts. Examples for such attacks are *XML injections* to manipulate the structure of messages or Denial of Service attacks via *oversized payload* using a very large message [18]. However, these attacks still pose a threat for cross-organizational SOA and have to be addressed by common countermeasures such as message validation and processing mechanisms [18]. This approach is depicted in Figure 3, using abstraction layers as an attack taxonomy for SOA.

Examples are service selection attacks, where differences in the security level a provider offers are exploited. The analysis of service consumer-provider-communication can also be an attack to gather information about the business (requests, used providers, time, frequency, etc.). Loosely coupled and malicious service compositions are a likely attack as well, if "good" services are encapsulated by "bad" ones in order to manipulate data or to gather information.

Compared to other work on SOA attacks [2,8,10,18] which have a strong focus on Web service technologies, this approach adds insights on attack scenarios which target specific SOA elements such as loose coupling and composability.

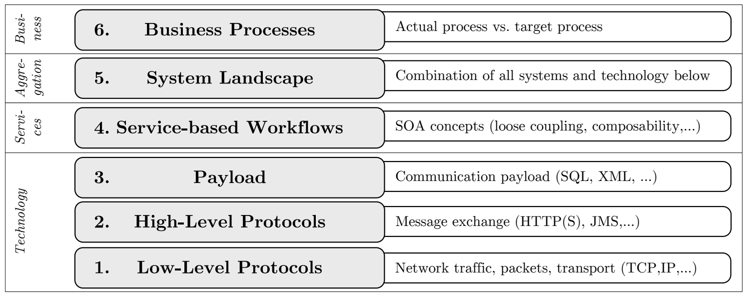


Figure 3. SOA security abstractions layers [14]

3.4 Selected Attack Defense Mechanisms

Based on the above results, several areas of interest for defense mechanisms were identified and now have to be evaluated regarding their potential for deeper research activities. These areas are the following:

1. decentralized service provider *reputation* for securing service compositions,
2. communication *obfuscation*, e. g., using Mixes [19], to avoid the exposure of information about business activities in the case of traffic analysis, and
3. service consumer *profiling* for detecting malicious service consumer behaviour.

It is planned not to pursue each of these areas in full depth, but to develop initial concepts in order to determine which area is the most fruitful for valuable contributions beyond existing approaches.

A prototype (`DFENS.KOM`) as basis for the above areas is already in development and it is based on an open-source SOA platform. There, service consumer and provider communication is relayed via a messaging system which was enhanced to forward messages to a proxy system. Via a configuration policy, this proxy is planned to trigger certain dedicated agents, e. g., for profiling consumer behaviour and checking against existing profiles, for obfuscating the communication to complicate traffic analysis, or for gathering reputation information about consumer and provider.

4 Conclusions and Future Work

As outlined above, the basic structure as well as the theoretical and conceptual foundation of the thesis is already in place (Sections 3.1–3.3) and will be further refined as follows:

- a critical revision of the conceptual SOA model elements regarding completeness, redundancies, and relationships,
- further extensions of the IT security metamodel regarding countermeasures and possibly adapting it to the Meta-Object Facility (MOF) standard,[2]
- identifying attack sub-steps from lower layers, the creation of a detailed attack model inventory, and implementing attack models for simulating attack behaviour in an Internet of Services scenario (`ATCK.KOM`).

[2] `http://www.omg.org/mof/`

However, the next major step will be to determine which of the above areas (reputation, obfuscation, profiling) to pursue further and in what direction. Basis for this will be an extensive review of related work, its applicability to the identified SOA-specific attack scenarios, and own initial solution concepts. Furthermore, appropriate evaluation techniques for such defense mechanisms have to be devised. This includes determining the evaluation basis, i. e., using a testbed based on our extensions of an open-source SOA platform or using simulation models for attacks and countermeasures in an Internet of Services scenario.

References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F., Krämer, B.J.: Service-Oriented Computing Research Roadmap. In: Dagstuhl Seminar Proceedings on Service Oriented Computing (SOC). (2006)
2. Josuttis, N.M.: SOA in Practice: The Art of Distributed System Design (Theory in Practice). O'Reilly Media, Inc. (8 2007)
3. Newcomer, E., Lomow, G.: Understanding SOA with Web Services (Independent Technology Guides). Addison-Wesley Professional (12 2004)
4. Schroth, C.: The Internet of Services: Global Industrialization of Information Intensive Services. In: Proceeding of IEEE ICDIM 2007. (2007) 635–642
5. Eckert, C.: IT-Sicherheit: Konzepte – Verfahren – Protokolle. 5th edn. Oldenbourg (11 2007) (in German).
6. Schneier, B.: Secrets and Lies: Digital Security in a Networked World. Wiley (2004)
7. Bishop, M.: Computer Security: Art and Science. Addison-Wesley (12 2002)
8. Kanneganti, R., Chodavarapu, P.: SOA Security. Manning Publications (1 2008)
9. Hafner, M., Breu, R.: Security Engineering for Service-Oriented Architectures. Springer, Berlin (2008)
10. Bundesamt für Sicherheit in der Informationstechnik: SOA-Security-Kompendium: Sicherheit in Service-orientierten Architekturen (2008) (in German).
11. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR (11 2004)
12. Melzer, I., et al.: Service-orientierte Architekturen mit Web Services. Konzepte – Standards – Praxis. 2nd edn. Spektrum Akademischer Verlag (2007) (in German).
13. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall PTR (8 2005)
14. Miede, A., Gottron, C., König, A., Nedyalkov, N., Repp, N., Steinmetz, R.: Cross-organizational Security in Distributed Systems. Technical Report KOM-TR-2009-01, Technische Universität Darmstadt (2009)
15. Hoglund, G., McGraw, G.: Exploiting Software: How to Break Code. Addison-Wesley (2 2004)
16. Barnum, S., Sethi, A.: Attack Patterns. "Build Security In" Initiative, National Cyber Security Division of the U.S. Department of Homeland Security (2006)
17. Schumacher, M.: Security Engineering with Patterns: Origins, Theoretical Models, and New Applications. 1 edn. Springer (9 2003)
18. Jensen, M., Gruschka, N., Herkenhoner, R., Luttenberger, N.: SOA and Web Services: New Technologies, New Standards – New Attacks. In: ECOWS '07: Proceedings of the Fifth European Conference on Web Services. (2007) 35–44
19. Raymond, J.F.: Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In: Designing Privacy-enhancing Technologies Workshop. (2001) 10–29

Knowledge-driven SOA Migration

Maryam Razavian

supervised by Patricia Lago
Department of Computer Science, VU University Amsterdam, the Netherlands
m.razavian@few.vu.nl

Abstract. Migration of legacy assets to SOA embodies a key software engineering challenge namely rehabilitation of pre-existing enterprise assets into service based systems. Existing service analysis and design methods mostly focus on development of new services while they lack in transforming services from already existing enterprise assets. We argue that, a comprehensive SOA migration methodology is essential for creating a well-constructed SOA out of pre-existing enterprise assets. To this end, the necessary knowledge concerning the migration should be identified, captured, analyzed and generalized. We propose an approach for migration that exploits relevant knowledge to transform an existing legacy system into a new service-enabled system. For this purpose, we will devise a reference knowledge model capturing types of knowledge that drive SOA migration.

1 Introduction

Facilitating the reuse of existing business functions from legacy systems is considered as one of the key features of the service oriented paradigm. Many organizations aim at modernization of legacy systems to SOA to achieve the advantages offered by SOA and still reuse the enterprise assets embedded in the pre-existing legacy systems. Since early use of SOA, migration of legacy assets to services has caught a lot of attention in research and industrial community. However, it is mostly assumed among researchers and developers that, existing enterprise assets are made to act as services simply by creating wrappers and leaving the underlying implementation untouched. It is worth mentioning that, the main goal of SOA is to promote highly standardized, loosely coupled services to foster easy composition of distributed applications [1]. Unless legacy elements are "inherently" suitable for use as a service with the aforementioned characteristics (which usually are not), it takes a considerable effort to provide an enterprise asset's functionality through services. As a result, a comprehensive SOA migration methodology is of critical importance for creating a well-constructed SOA out of pre-existing enterprise assets.

Let us consider the following example:

A pre-existing legacy system stores and manages a patient's medical records. The existing knowledge about this legacy system includes the description of the provided functionality, the business processes the system supports (e.g. update medical record, analyze patient history to formulate a diagnosis), and the business value for the current/potential

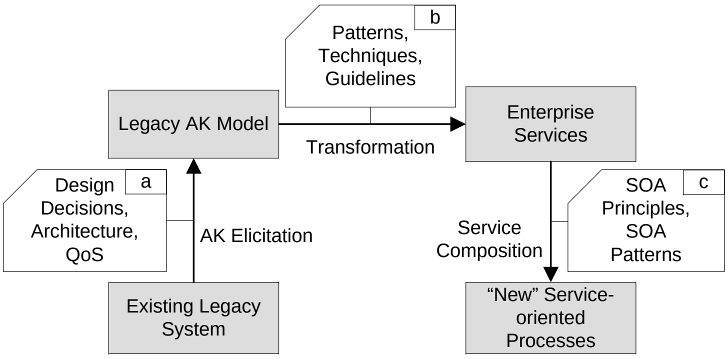


Fig. 1. SOA Migration Process

users (like the medical doctor) embedded in the legacy system. Such information can then be translated into business processes, business data and user added-value, which in turn can be associated with, respectively, task-centred services, data-centred services and end-user applications. For instance, if the knowledge regarding business data and functionalities associated to patients is not extracted, a smooth transition to patient data service is not possible. All the knowledge concerning the existing patient information system, the target SOA environment (e.g. types of services) and transformation techniques (e.g. wrapping) affect the migration process in terms of "what"is going to be migrated and "how" the migration is performed.

Despite its simplicity, this example already reflects essential requirements and challenging issues of the migration methodology from a knowledge management perspective:

- The main types of knowledge concerning the pre-existing legacy system should be identified (Box a, Fig.1)
- The knowledge concerning elements of service based systems should be extracted (Box c)
- Methods and guidelines should be devised for transforming legacy elements into types of services (Box b)

We argue that a comprehensive knowledge management approach is of great importance for SOA migration methodology. The reason behind this is that, knowledge management helps to rationalize the investigation of legacy assets as candidate services, isolate their properties and transform them into meaningful business services while adhering to service relevant aspects and challenges. This research is part of a project named the Service-enAbling PreexIsting ENterprISe Assets- Methodology (SAPIENSA) aiming at proposing a migration methodology for creating a well-constructed SOA out of pre-existing enterprise assets. Our contribution in this project is to effectively exploit architectural knowledge to migrate an existing enterprise system into a new service-enabled system.

2 Proposed Solution

In this section we further explain our proposed solution. To clarify the idea, we recall the example presented in Section 1. The following types of knowledge could shape the migration process of the medical information system: 1) the body of knowledge concerning the properties of existing system including lost abstractions (structural design, business processes, business rules, etc.) and know-how (design decisions, discarded alternatives, etc.) 2) patterns, templates, and methods that address altering and reshaping the extracted legacy elements to relevant type of services 3) SOA specific characteristics and properties. We aim at devising a reference knowledge model capturing types of knowledge that drive SOA migration. To this end, we propose to analyze different migration processes (from industrial cases) and extract the implicit and explicit types of knowledge, which shape the migration. We propose the phases depicted in Fig. 1 for the service-enabling methodology using architectural knowledge (AK). We argue that these phases facilitate extraction of a reference AK model as it helps gathering architectural insights and transforms them to well-constructed services. These phases along with their associated research challenges are discussed in the followings.

2.1 Architectural Knowledge Elicitation

The main goal of this phase is to identify the legacy elements that are relevant for migration to SOA and make the decisions regarding how to transfer to services. To this end, we aim at identifying, isolating and describing the knowledge about the typical "standard" elements that belong to pre-existing enterprise assets, and generalize the reusable knowledge. In order to address these aspects, the following research questions should be resolved:

– *RQ1) what are the knowledge elements that are relevant to be made explicit?* For instance, solution-related knowledge such as architecture, design decisions, etc., as well as problem-related knowledge including functional requirements (e.g. business processes and business rules) and non-functional requirements (e.g. privacy and security), should be externalized.

In order to enable an effective knowledge capturing and sharing, the knowledge extracted in this phase should be explicitly represented. Architectural knowledge models aiming at representing decisions in general are considered as a potential means of codifying the elicited knowledge. For instance, the AK core model developed in our previous work [2] is relevant for making AK explicit. Knowledge elements specific to an organization or an application domain should be defined as specific extensions to the core model described above. We consider service-oriented enterprise architectures as SOA extensions.

2.2 Transformation

The transformation phase embraces the actual migration of legacy assets to service based assets. The AK acquired during the AK elicitation phase will be used as a fabric for creating new business services. This transformation could be in the form of reshaping design elements, restructuring the architecture and/or altering business models and business strategies. It should be noted that, each one of these forms of transformation belongs to a specific level of abstraction. For instance, a legacy element can be transformed to services by altering its encapsulation using wrapping techniques. In the same way, a composition of legacy elements can be transformed to a service or composition of them. At a higher level, an existing business model is transformed to a to-be business model based on new requirements as well as opportunities offered by service based systems. In all these cases the knowledge element extracted within AK elicitation phase is converted to another knowledge element in the SOA environment. Feasibility of each type of migration depends on many additional factors, such as past design decisions and knowledge about extra functional aspects like technological constraints or quality attributes. It becomes therefore even more important to capture the relevant AK in pre-existing and new systems in order to assess the feasibility of different transformation techniques. Recall that in this research work, we aim at isolating and describing the AK about the typical "standard" elements that belong to pre-existing enterprise assets; in the same way, we want to describe the AK about the "standard" elements of an SOA, including the architectural patterns relevant for SOA transformations like for instance service topologies and worst-case quality concerns. By carrying out a number of industrial case studies, we will define how we can typically transform a standard element of preexisting enterprise assets, into a "standard" element of an SOA. This will result in a number of transformations, as represented in the horizontal, continuous arrow in Figure 1. In this way, the "SOA migration decision making process" will be supported by a knowledge base of SOA transformations capturing necessary technical and nontechnical AK. In order to address all these aspects of transformation, following research questions should be addressed:

- *RQ2) what are the elements which drive the transformations?* For instance, past design decisions, quality attributes or technical constraints can have influence on the solution regarding the transformation problem.
- *RQ3) what are the possible types of transformation regarding different levels of abstraction?*
- *RQ4) how to represent the body of knowledge associated to transformation?*

2.3 Service Composition

Processes need to be designed in terms of constellations of the interacting services obtained in the transformation phase. This is supported in the service composition phase, which provides the ability to restructure, compose and decompose services. Certainly, bodies of knowledge addressing SOA environment, service association, message partnership, transaction ownership, etc., have to be captured. This phase is researched by another partner of the SAPIENSA project.

3 Related Work

Our work aims at enabling the migration process with knowledge capturing, sharing and reusing. Therefore, we position ourselves in the fields of architectural knowledge management and migration process. Accordingly, we discuss the related work in these two research fields. A vast body of work in the area of SOA migration mostly focuses on exposing legacy code as (web) services. Sneed proposes an approach that includes salvaging legacy code, wrapping the extracted code and making the code available as services [3]. Typically, the focus of these works is limited to implementation aspects of migration which usually covers techniques to alter a segment of legacy code to web services. A further family of approaches aims for covering the whole migration process. These approaches are comprised of two main sub-processes: top-down service development and bottom-up service extraction [4, 5]. The main goal of the bottom-up sub process is to support understanding of existing legacy system and extracting services from legacy code using code analysis and architectural recovery techniques. Within the top-Down process the target services are designed and implemented based on existing legacy components as well as new requirements in the problem domain.

Recently, there is an increasing focus on management of architectural knowledge including architectural design decisions and their rationale [6, 2]. However existing work focuses on capturing and representing the AK as such it is relevant for any application domain. To the best of our knowledge, there are limited results in identifying the architectural knowledge relevant for SOA. A specific type of AK is contained in architectural and design patterns, which document standard solutions to recurring problems in software design. Some research is emerging in identifying SOA patterns [7, 8], addressing several important aspects of service development and management. In these approaches, disciplined design decisions and compliance to SOA patterns are considered key criteria for successful development. Still, the documented knowledge mostly captures reusable technical solutions, while the related rationale or decision process that led to the solution is missing.

4 Research Plan

This research is carried out in a consortium of two universities and industrial partners. These industries provide us with case studies and give regular feedback. My research work will adopt an iterative incremental approach where the industrial partners provide industrial cases on migration of legacy systems to SOA. The main goal here is to identify the state of practice regarding the migration problem. More specifically, we are seeking to find out how migration process is performed in practice.

Also, we aim at identifying the kinds of activities that are performed and the types of knowledge that are captured. To this end, we plan to perform the following tasks:

- Elicit AK of pre-existing enterprise assets from case studies and model it as SOA extensions to the AK core model (RQ1)
- Devise from case studies the transformations to candidate services, SOA patterns and related AK (RQ2, RQ3, RQ4)
- Integrate candidate services into the target service-oriented system

After a startup phase defining the methodology to be used and the AK reference model for pre-existing systems and SOA, three iterations will occur (one for each project year). In each iteration, the cases carried out will be followed by a reflection period aimed at giving feedback on migration process and AK reference model, and defining the possible transformation methods. A final phase will consolidate all results in a generalized and reusable solution. The expected contributions of this research work are as follows:

- SOA model representing the AK relevant for SOA migration
- Patterns codifying architectural practices as ready-to-use solutions for the migration problem
- Prototypes and test beds supporting the reference knowledge model for SOA, including a pattern repository, a visual modeling environment and AK search

5 Acknowledgement

This research has been partially sponsored by the Dutch Joint Academic and Commercial Quality Research and Development (Jacquard) program on Software Engineering Research via contract 638.001.206 SAPIENSA: Service-enAbling PreexIsting ENterprISe Assets; and the European Community's Seventh Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

References

1. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal **16**(3) (2007) 389–415
2. de Boer, R.C., Farenhorst, R., Lago, P., van Vliet, H., Clerc, V., Jansen, A.: Architectural knowledge: Getting to the core. In: QoSA. (2007) 197–214
3. Sneed, H.M.: Integrating legacy software into a service oriented architecture. In: CSMR '06: Proceedings of the Conference on Software Maintenance and Reengineering, Washington, DC, USA, IEEE Computer Society (2006) 3–14
4. Zhang, Z., Yang, H.: Incubating services in legacy systems for architectural migration. In: APSEC '04: Proceedings of the 11th Asia-Pacific Software Engineering Conference, Washington, DC, USA, IEEE Computer Society (2004) 196–203
5. Andreas Winter, J.Z.: Model-based migration to service-oriented architecture. In: the International Workshop on SOA Maintenance Evolution (SOAM 2007), 11th European Conference on Software Maintenance and Reengineering (CSMR 2007). (2007)
6. Tyree, J., Akerman, A.: Architecture decisions: Demystifying architecture. IEEE Softw. **22**(2) (2005) 19–27
7. Zimmermann, O., Grundler, J., Tai, S., Leymann, F.: Architectural decisions and patterns for transactional workflows in SOA. (2007) 81–93
8. Erl, T.: SOA Design Patterns. Prentice Hall PTR, Upper Saddle River, NJ, USA (2009)

Application of Data Mining to Performance Management of Distributed Enterprise Systems

Sergio Pacheco-Sanchez[1,2],
supervised by {Bryan Scotney[2], Sally McClean[2], and Gerard Parr[2]}

[1] SAP Research CEC Belfast,
Building 1, The Concourse, Queen's Road,
Belfast BT3 9DT, U.K.
sergio.pacheco-sanchez@sap.com
[2] School of Computing and Information Engineering,
University of Ulster, Cromore Road,
Coleraine BT52 1SA, U.K.
{bw.scotney, si.mcclean, gp.parr}@ulster.ac.uk

1 Introduction

The performance evaluation of a computer system involves constructing a suitable model of the system and then using it to predict the system behavior. The model incorporates domain-specific and a priori knowledge about the system inner structure and the workload. The model may be solved either by using analytical techniques or simulation or both. The fundamental parameters needed to specify performance models are arrival times and service demands. The former refers to the frequency of new incoming requests entering the system. The latter indicates the amount of time a request needs to be served, therefore its precise estimation is of vital importance for defining models that are both representative and robust. The number of resource consumption estimation methods that can provide an adequate estimation of the service demand placed on the system [7, 9] is limited compared to the various modeling and evaluation techniques that exist to obtain performance predictions. Essentially, existing techniques for characterization of resource consumption needed for model parameterization are based on direct data measurement and statistical inference. Direct data measurement techniques propose to set up performance probes in the system possibly at different levels, depending on the layers to be analyzed. For example, Magpie [1] automatically extracts system workload characteristics using low-overhead instrumentation. It correlates system generated events to the control flow and to the resource consumption of the requests. Although effective, it is not always possible to adopt instrumentation based approaches of this type in production systems, thus statistical inference is often preferred. Statistical inference encompasses the use of statistics and random sampling to make inferences concerning some unknown aspect of a given population, e.g. response time of service requests. For instance, statistical learning techniques are appealing because they may assist in building system models or subsets of them in an automated or semi-automated manner with limited previous knowledge of the system environment. The present work proposes to investigate the way: (i) machine learning can be used in workload characterization tasks; (ii) statistical inference can assist in the correlation of low-level metrics for prediction of system states and in the characterization of resource consumption for system analysis and model parameterization purposes. Initial work done in the latter presented two methods for estimating the service demands of requests based on measurement of their response times [6]. This approach is particularly suitable in scenarios of virtualized systems or for services controlled by third parties. Not surprisingly, the impact of this approach may be of relevance in the near-future, especially when it is forecasted that by 2015 more than 75% of Information Technology infrastructure will be sold as a service by service and infrastructure providers [3]. Preliminary results show the very good accuracy of the proposed approach and a further extension of this investigation is presented in Subsection 3.1. The remainder of the paper is organized as follows. The research challenges of the investigation are presented in the following Subsection. Related work is then introduced in Section 2, followed by a detailed description of the research proposal in Section 3, including (i) and (ii).

1.1 Research Challenges

This subsection offers a short glimpse into the open research fields of computer performance evaluation, namely parameterization of performance models, workload characterization and correlation.

Parameterization of performance models. The challenge of model parameterization is to determine the mean service demand $E[D]$ of the incoming requests at the server such that the response times predicted by the model match accurately those measured in the real system for all possible numbers of users. The predominant approach to model parameterization makes use of explicit system performance models, such as control-theoretic or queuing-theoretic models. These approaches have achieved major success in many specific performance management applications. However, it is noted that the development of accurate models of complex computing systems is both complicated and highly knowledge-intensive, and moreover, such modeling should become progressively more difficult as systems become increasingly complex and distributed.

Workload characterization. A system workload may be composed of several types of transactions or requests; each one may stress a system in a different manner, therefore possibly consuming system resources differently. Every transaction or request type is also known as class. A key challenge in workload characterization is to determine in an automated manner the number of workload classes. For instance, the work done in [8] embraces such a challenge by proposing an inference-based technique to solve the request categorization problem using as input only high level aggregated system measurements (e.g. CPU). However, more work needs to be considered in the area for exploring the limits and scalability of similar approaches.

Correlation. Another open question in this field is: How do we know what system-level metrics are 'Key Performance Indicators' (KPIs) of a system?. In other words, it is important to understand the relationship between low level metrics and system states for achieving system performance management excellence. It is preferred to have small subsets of metrics that capture most of the patterns of performance behavior in a way that is accurate and helps to explain the causes of observed performance effects.

2 Related Work

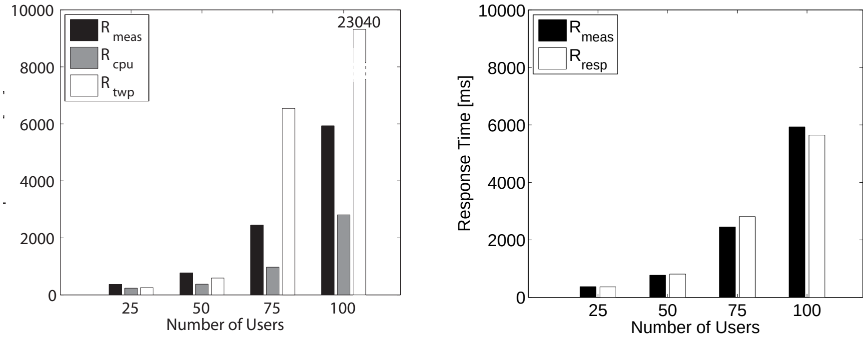
There are two main trends in the literature regarding system modeling for distributed systems. One trend consists of analytical models based on queuing networks [9], control theory [4] and Petri nets [5]. Although mature and mathematically sound, most of the aforementioned models have several limitations: i) the models might be complicated to build, for example, due to absence of knowledge about black-box components; ii) model construction can be prone to human errors; iii) models built are a priori and might not adjust properly to workload or system changes; iv) assumptions that diverge considerably from actual system conditions may have to be made. The second trend is the emerging research rooted in statistical learning which traditionally have used general-purpose learning algorithms. However, in a practical problem where there is prior knowledge it can be difficult to incorporate such knowledge into these very general algorithms. Moreover, the classifiers built by these general algorithms are often difficult to interpret as their inner structure may not reflect the real-world process being modeled. An approach to solve this problem has been the use of generative stochastic models. A stochastic model is a model that describes the real-world process by which the observed data is generated. By and large, the process of learning a stochastic model consists of the following steps: a) choosing a graph structure; b) specifying the form of the probability distribution at each node in the graph; c) fitting the parameters of those probability distributions to the training data. In most applications, steps a) and b) are performed by the user and step c) is performed by a learning algorithm. However, recent research has made progress in developing algorithms for learning the structure from data as well. For example, Tree-Augmented Bayesian networks (TANs) are learned from data to correlate low level system measurements, e.g. CPU usage, etc., with system states [2]. Thus, once a stochastic model has been learned, probabilistic inference can be carried out to support tasks such as classification, diagnosis and prediction.

3 Research Proposal

Developing effective models for performance management in distributed computing systems is an important goal of current systems research. Heading in this direction, the present work aims to investigate: i) mechanisms to improve the process of model parameterization, essential to yield reliable performance predictions; ii) correlation of low-level metrics for prediction of system states; iii) automatic or semi-automatic mechanisms for workload characterization. At this stage, preliminary work has been done using a simple *M/M/1* queuing model [6], however, it is needed to further investigate whether other models can capture system performance more accurately. The discovery of hidden associations between low level measurements and the manner by which they impact the final perception of performance may yield in a deeper understanding of the system behavior. Both the mechanisms for model parameterization and workload characterization can benefit from this outcome.

3.1 Estimating Service Resource Consumption

In the current work, it is proposed to conduct further model investigation and experimental studies for estimating service resource consumption. This is the extension of previous work [6] that for the first time elaborates the estimation of service demands without knowledge of server utilization measurements. Instead, linear regression (RR) and maximum likelihood (ML) methods are utilized for estimating the request service demands based on the response time measurements only. This means server instrumentation or sampling is not required, therefore the proposed approach significantly simplifies parameterization of performance models when measuring utilization is difficult or unreliable, such as in virtualized systems (because filtering hypervisor system overheads is troublesome and rarely done) or for services controlled by third parties. Furthermore, an intrinsic advantage of the proposed approach is that response times at a server depend on all the latency degradations incurred by requests within the system, therefore they are inherently more comprehensive descriptors of performance as they also account for bandwidth, memory, and I/O contention delays. Very rarely, these components are all directly accounted in models, yet they can be critical performance drivers and are ignored by the utilization approach. During the experimentation phase the RR method service demand estimates (obtained by computing the linear regression of a sequence of response time samples and arrival queue-length averaged on 20 consecutive requests) were used to parameterize a closed *M/M/1* queue model representing an industrial ERP application. Figure 1(b) shows that the response time predictions obtained through the RR method (R_{resp}) match closely the real response time measurements (R_{meas}) by achieving an accuracy of 90-95%. In contrast, the response time predictions provided by the utilization based approaches, namely R_{cpu} and R_{twp}, clearly underestimate (on average by 50%) and overestimate (on average by 130%) R_{meas} respectively, as depicted in Figure 1(a). R_{cpu} is obtained from the estimate of $E[D]$ using linear regression of the cumulative times for CPU and DB processing against the number of completed requests, whereas R_{twp} is obtained from the estimate of $E[D]$ using linear regression of the total time spent in the work process after admission in the system against the number of completed requests. The learning from this early work reveals that further sensitivity analyses need to be done by considering various aspects. For example, the impact of non-exponential features in arrivals



(a) Utilization approach (b) Response time approach

Fig. 1. Comparison of response time predictions.

and service demands for the different request classes and utilization levels is unknown. Regarding the queue-length seen on arrival to the system, it is important to investigate the outcome of using a steady state distribution instead. Another aspect requiring investigation is the utilization of the *M/GI/1/PS* queue. Finally, it is crucial to understand how the lack of information on the residual time of requests impacts on the overall accuracy.

3.2 Statistical Correlation

It is planned to use TAN models to represent the relationship between the low level metrics and the high-level system state as a joint probability distribution. Out of this distribution, a characterization of each metric and its impact on the system state, e.g. violation of a quality of service contract agreed as a maximum two-second response time, can be extracted. The measurements collected from the ERP case study used in subsection 3.1 are vast. It includes several dozen attributes such as sequential read database (DB) rows, number of DB procedure calls, DB update average time/row, time spent in the work process, wait time, CPU time consumed, etc. However, it is important to consider the fact that smaller subsets of metrics allow a more efficient representation and evaluation of the model.

3.3 Machine Learning

The idea of considering the request categorization problem as a Blind Source Separation (BSS) problem seems to be novel as presented in [8]. BSS aims to separate a set of signals (or vectors of measurements) from a set of mixed signals (or complex workload), without the aid of information or with very little information about the source signals or the mixing process. Those vectors of measurements are grouped in request categories, which represent classes of requests that have similar resource consumption requirements. Therefore, inferring various characteristics of a workload has direct application to the performance modeling of an application, in other words it simplifies the performance model parameterization process. The work done in [8] proposes to use a machine learning technique known as Independent Component Analysis (ICA) to solve the underlying BSS problem. It presents preliminary results on a synthetic workload using an e-commerce benchmark, showing an error within 4 to 17%. Inspired by the potential of such work, it is proposed to investigate the application of similar data-driven techniques for BSS including ICA, Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) in a real ERP environment. In contrast to [8], the current investigation will use representative workloads that include standard business transactions such as creating and listing sales orders and invoices. Similar to the investigation conducted in [6], it is planned to use a realistic workload model with several classes and conclude whether the techniques are indeed scalable, given the fact that they were used in [8] within an experimental study based on an overly simple workload model.

4 Acknowledgement

This research has been co-funded by the Northern Ireland government within the InvestNI/SAP program. The author thanks to the MORE project colleagues Giuliano Casale, Stephen Dawson and Stephan Kraft and also to Franklin and Andrea Jones who greatly helped in improving the quality of this paper.

References

1. P. Barham, A. Donnelly, R. Isaacs, and R. Mortier. Using magpie for request extraction and workload modelling. In *OSDI'04*, CA, USA, 2004. USENIX.
2. I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J. S. Chase. Correlating instrumentation data to system states: a building block for automated diagnosis and control. In *OSDI'04*, pages 16–16, CA, USA, 2004. USENIX.
3. S. Gaisbauer, J. Kirschnick, N. Edwards, and J. Rolia. Vats: Virtualized-aware automated test service. In *QEST '08*, pages 93–102, 2008.
4. E. Kalyvianaki, T. Charalambous, and S. Hand. Self-adaptive and self-configured cpu resource provisioning for virtualized servers using kalman filters. In *ICAC '09*, pages 117–126. ACM. NY, USA, 2009.
5. S. Kounev and C. Dutz. QPME - A Performance Modeling Tool Based on Queueing Petri Nets. *ACM SIGMETRICS Performance Evaluation Review*, 2009.
6. S. Kraft, S. Pacheco-Sanchez, G. Casale, and S. Dawson. Estimating service resource consumption from response time measurements. In *ValueTools '09*, ICST, Brussels, Belgium, to appear in 2009.
7. J. Rolia and V. Vetland. Correlating resource demand information with arm data for application services. In *WOSP '98*, pages 219–230. ACM. NY, USA, 1998.
8. A. B. Sharma, R. Bhagwan, M. Choudhury, L. Golubchik, R. Govindan, and G. M. Voelker. Automatic request categorization in internet services. In *HotMETRICS'08*. ACM. NY, USA, 2008.
9. Q. Zhang, L. Cherkasova, and E. Smirni. A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In *ICAC '07*, DC, USA, 2007. IEEE.

Maintenance of customized processes

Pavel Bulanov*
supervised by Marco Aiello*

{p.bulanov,m.aiello}@rug.nl
University of Groningen
Nijenborgh 9, 9747AG Groningen,
The Netherlands

Abstract. When similar business processes are deployed in different organizations, one typically adapts manually a template process to the different execution environments and organizational needs. Service-Oriented Architectures facilitate the task of adaptation by exposing the functionalities needed by the process in an abstract way, decoupled from the implementation. Nevertheless, manual intervention to customize the processes is still highly necessary, making it hard to maintain the processes obtained as the result of customizations. Even more complex is the situation where several processes are customized independently but they are still treated as linked with each other or inherited from one base (reference) process.
In this paper, we provide an initial investigation of customization of controllable processes.

Keywords: Variability management, Customization, eGovernment

1 Introduction

The idea of having one process to fit all needs is not realistic. In general, one either designs a process for a very specific task, or designs a template of a process more or less formally, to be adapted by different parties to their specific situation. Governmental laws are a good example of such circumstances. The central government approves a law, which is a collection of requirements expressed in a peculiar type of natural language. The law is then accompanied by an interpretation document which explains how to implement it, again expressed in natural language. Then the bodies involved in the law need to take action in order to manage/enforce/adapt to the law's indications.

The service-oriented approach, together with initiatives on corresponding standardization and implementation are strongly pursued by the Dutch government. In this work we deal with more technical aspects of these initiatives. We concentrate on the area of workflow management, which has seen great progress in the last years (e.g., [1]) and, if we consider the brand of Service-oriented architectures known as Web services [5], the natural candidate for modeling processes is the Business Process Execution Language (BPEL) [2].

* Distributed Systems Group

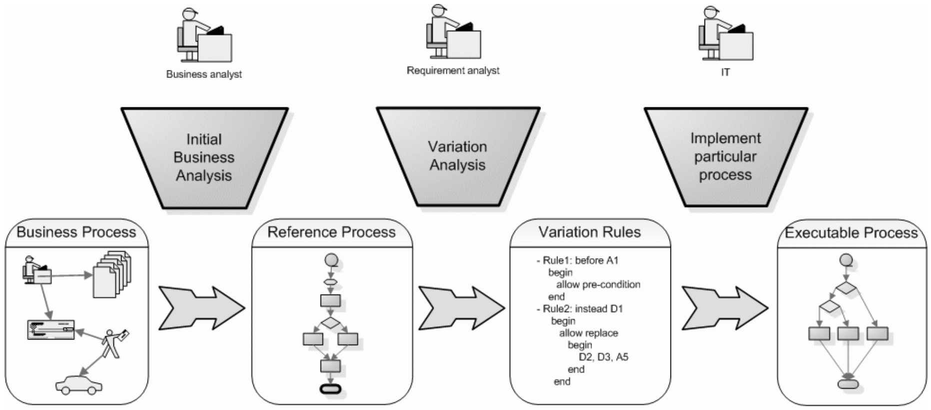


Fig. 1. Overview of the two-phase transformation.

However, BPEL has some limitations with respect to the vision of process customization. BPEL represents a business process as a tree-like structure with the individual activities as nodes, and the control of flow statements (if-else, while etc) as branching points within this tree. If the process itself is complex then this tree will also be complex and hard to understand by a human reader. Yet more problems rise while the process needs to be customized. There are several extensions like [3] or [4] which introduce a way to simplify representation of the complex business process via dividing it into the two parts — the “core” process and the additional rules over it. It simplifies the understanding of the whole process while providing a way to make customizations in the form of additional rules.

In case of a regulated environment while there is a need in keeping control over the possible customizations, those extensions will not help since they do not provide an explicit control over customization process. Such a control should be performed manually and there is no guarantee the changes applied are compatible with the existing regulation constraints.

To manage customization in a regulated environment, such as that of eGovernment, we propose a process language that allows us to express variations and constraints, and also a two phase transformation framework to manage the customization and maintenance of the processes. The overall transformation is illustrated in Figure 1. On the left, one starts with a description of a process, perhaps informal. The business analyst then transforms this into an unambiguous process description and then adds variations and constraints to make sure that the process described is as widely applicable as possible. This concludes phase 1, the output of which is a process with variations described and a set of constraints to be satisfied. Phase 2 consists of the customization of the process at the many different points of execution. This last step is fully automatize and initiated by an IT specialist.

2 Issues with process maintenance

The approach proposed in this article makes it easier to maintain process evolution. For example, consider a situation in which a reference process is changed due to the fixing of an error or a change in the requirements for implementation. In this case, it is required to repeat the second phase of transformation in order to propagate the changes in all inherited process automatically. Such an advantage is achieved via the restriction of possible customizations as well as dividing the customization into the two formal phases.

On the other hand, while predictable change can be managed in this way, it is impossible to anticipate all the possible changes in the requirements. Therefore, this solution does not work for all cases, and this is why the solution is offered for regulated environments such as eGovernment where it is easier to restrict the possible changes by means of administrative rules. Also, under some circumstances it is necessary to change the inherited process directly due to local needs which were not anticipated by the original design. The whole picture is illustrated in Figure 2. Here the reference and inherited processes are displayed, and the horizontal arrow labeled with “1” represents the existing transformation rules which are forming the link between the reference and inherited processes.

The reference process may also be changed, leading to athe new reference process; this is illustrated by the left vertical arrow labeled with “2”. As a result, having the link between the reference and inherited process, it is possible to build a new inherited process via repeating the underlying transformation. Obviously, this transformation may become invalid over the changed reference process and thus the first issue appears: *How can we identify whether the changes being applied to the reference process are compatible with all inherited processes?*

At the same time, the inherited process may also be changed, which is reflected by the right vertical arrow labeled with “4”, and in that case the second issue appears: *Once the inherited process has changed, is it feasible to keep the link with the reference process?* In other words, is it possible to maintain the link between the inherited process and the base process, while at the same time retaining the ability to make arbitrary modifications within the inherited process?

Note that the transformations “1” and “3” are the same since they both represent the link between the reference and inherited process. The remaining problem is that the transformation “3” may be invalid, as already described as the issue one.

3 Ways to address the issues

Several approaches may be identified in order to solve aforementioned issues related to the workflow maintenance and evolution:

- Make an automatic merge of two workflows basing on formal graph transformation rules
- Confine the possible customization changes so it would be easy to perform a maintenance — the “difficult” cases in customization just not allowed

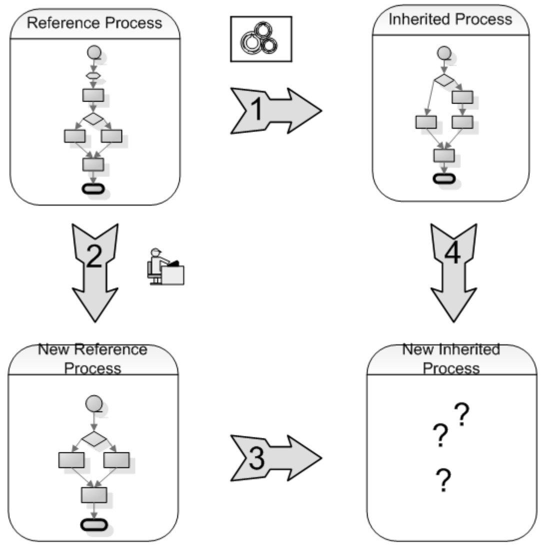


Fig. 2. The process maintenance.

– Identify a “change set” or “change region” and treat it like a base building block of customization.

The first approach requires complex analysis of the merged graph since apart from the geometric difference we also have to consider semantic meanings for each node as well as the side effects of any change in the process structure. These issues represent a challenge which is beyond the scope of this paper.

The second approach is the easiest one but reduces the flexibility, thus making it difficult to use for real world tasks. This approach is based on the assumption that the particular “difficult” customization cases are not important and may be suppressed. Such an assumption could only be made after analysis of the particular domain area and identification the most frequent customization cases. Therefore, is approach is not suitable for a general-purpose solution.

For the third approach there are many investigations related to the identifying of a change set, for example [6]. The extraction of a change region may help to construct a conversion function as an interpretation of manual changes. This conversion function corresponds either the arrow “4” or the arrow “2” on Figure 2, depending on which process is changed. Once the conversion function is extracted it is possible to use the combination with the existing transformation function — either the combination of “2” and “3” (when the reference process was changed) or “1” and “4” (when the inherited process was changed).

Regarding the third approach, many investigations have been carried out related to the identification of a change set, for example [6]. The extraction of a change region may help to construct a conversion function as an interpretation of manual changes. This conversion function corresponds either to the arrow “4” or the arrow "2" in Figure 2, depending on which process has changed. Once the conversion function has been extracted, it is possible to use the combination with the existing transformation function - either the combination of "2" and "3" (if the reference process has changed) or "1" and "4" (if the inherited process has changed).

The main issue in relation to this approach is to extract the conversion function correctly and to identify the possible cross-influence between this extracted conversion function and the existing transformation link. This is a subject for future research.

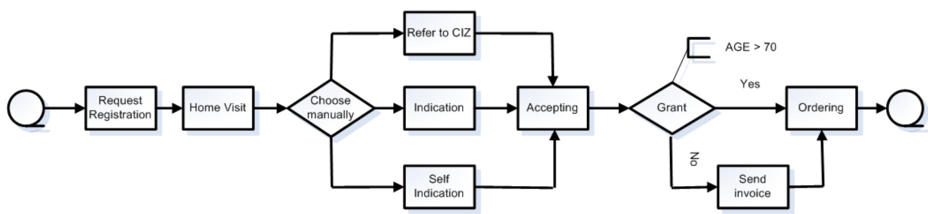


Fig. 3. An example of an e-Government process for obtaining a wheelchair.

Consider an example of a typical eGovernment process — issuing a wheelchair for elderly people (see Figure 3). Reading the figure from left to right, we observe the initial activity of registering to obtain a subsidized wheelchair. The next activity is that of an expert from the municipality visiting the home of the requesting person. Then there is a choice point requiring a choice being made by an authorized civil servant. Based on this choice, three further activities are possible, which then rejoin in an accepting the request activity. Now a further automatic choice is performed, such as a check on the age of the requesting party.

Finally, the original person will receive a wheelchair, either having to pay for it himself, or having it subsidized. However, in one municipality the minimum age for getting a wheelchair may be 70, and in another one it may depend not only on the age.

Now, consider the case in which the reference process is not acceptable for one of the local municipalities, so they make an appropriate change and as a result is the process shown in Figure 4. The changed regions are outlined with dotted rectangles for convenience.

What happens if the reference process itself is changed after the customization has been done? For example, the reference process is modified in a way when the activity "Refer to CIZ" is no longer needed . This modification is not dangerous since the activity which is removed from the reference process is not used in the inherited processeither. On the other hand, this activity belongs to the *change region* which may lead to difficulties with automatic resolving.

Another example of the reference process change is the removal of the activity "Indication", which not only belongs to the *change region* but is also used by the inherited process. Two options are available here: to remove this activity from

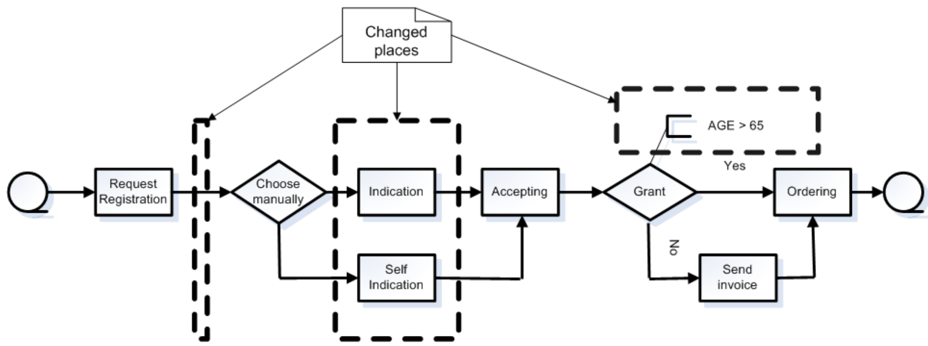


Fig. 4. Customization example

the inherited process as well or to report an error since such change is illegal. It is not possible to identify which option is correct since the decision depends on the logic beyond the processes and the nature of changes made to those processes.

These simple examples show the difficulties with the automatic process maintenance therefore making the task of automatic process maintenance challenging.

Acknowledgements

The research presented in this paper is supported by the NWO Jacquard Software Engineering program with the project Software as a Service for the varying needs of Local eGovernment (**SaS-LeG**), with partners: the University of Groningen, Cordys and an association of Dutch Municipalities. Contract no. 638.000.000.07N07. Project website: http://www.sas-leg.net.

References

1. W. van der Aalst and K. van Hee. *Workflow Management*. MIT Press, 2002.
2. BPEL4WS. *Business Process Execution Language for Web Services*, May 2003.
3. Anis Charfi and Mira Mezini. Hybrid web service composition: business processes meet business rules. In *ICSOC*, pages 30–38, 2004.
4. M. Momotko, E. Tambouris, G. Blizniuk, W. Izdebski, and K. Tarabanis. Towards implementation of life-events using generic workflows. In *eGovernment Workshop '06 (eGOV06), Hosted at Brunel University*, 2006.
5. Mike P. Papazoglou and Willem-Jan van den Heuvel. Service oriented architectures: approaches, technologies and research issues. *VLDB J.*, 16(3):389–415, 2007.
6. Wil M. P. van der Aalst. Exterminating the dynamic change bug: A concrete approach to support workflow change. *Information Systems Frontiers*, 3(3):297–317, 2001.

Towards an Abstraction Layer for Scientific Services

Marcos Baez

Supervised by Fabio Casati
University of Trento – Dipartimento di Ingegneria e Scienza dell'Informazione
Via Sommarive 14, 38100 Trento, Italy
baez@disi.unitn.it

Abstract. This paper presents a platform that facilitates building scientific services on top. We describe the problems in building such services and derive a general-purpose, extensible layer for accessing any resource that has a URI and is accessible on the Web. The platform, and more in general the classes of systems that have this functionality, is referred to as Resource Space Management System and is the (scientific) resource analogous of a data space management system. In this paper we describe the model and conceptual architecture of the platform, discuss its benefits and outline the research plan for its realization.

Keywords: Resource Space Management Systems, Scientific Resources, Scientific Services

1 Introduction

With the advent of the Web era we have moved away from printed papers and journals towards digital formats. As a result, a large number of services allowing their dissemination, archiving, sharing and reviewing have emerged. This has also made possible the rising of other non-conventional types of *scientific contributions*, e.g., video, datasets, and other resources; very rare before the Web. Thus, the Web has opened a world of possibilities for how scientific knowledge dissemination, creation and evaluation could be done and for how the notion of scientific contribution could evolve to serve the need of scientists to learn about novel, interesting research ideas and results.

There has been a considerable amount of work, ranging from theoretical to practical proposals, on how to exploit the Web to improve the way we do science today. Perhaps, one of the most representative of such initiatives, due to is ambitious goals and scope, is *Liquidpub*[1], an EU project that aims at capturing the lessons learned and opportunities provided by the Web and open source, agile software development to develop concepts, models, metrics, and tools for an efficient (for people), effective (for science), and sustainable (for publishers and the community) way of creating, disseminating, evaluating, and consuming scientific knowledge [1].

Besides concepts and models, from a technological/service perspective, imple-

[1] http://project.liquidpub.org

menting the vision of initiatives such as Liquidpub requires the development of *services* for, among others, i) supporting scientists in knowledge search, aggregation, and evaluation by interfacing with "traditional" or "novel" data sources (from SpringerLink and ACM repositories for access to scientific data and metadata to social bookmarking sites such as citeUlike), ii) interfacing with systems that support the knowledge evaluation process (e.g., conference management systems), and iii) interfacing with systems that provide for early sharing of knowledge (blogs, wikis). These are just some examples of the arbitrary number of scientific services that can be built once access to scientific resources is available.

Given the above, we need an extensible and common platform to access the various kinds of scientific resources available on the Web, that makes it easy (or at least easier) to develop services on top. The goal of this research work is to design and develop such a platform; which we refer to as a *resource space management system* (RSMS). We plan to achieve this by providing the RSMS with the following characteristics:

- **Homogenous programmatic access to scientific resources and web services** regardless of how they are implemented as long as they are web accessible (via browser or rest/soap API).
- **Universality**. We aim at covering a large set of scientific resources of various kinds as described above. While anything identified by a URI is in scope of RSMS (whether it is a scientific resource or not) we aim to provide concepts and services for scientific resources, such as built-in notions of authors, references, and the like.
- **Collaborative Extensibility**. Given the large amount of services available, it is practically impossible to provide a monolithic infrastructure that incorporates all of them. We made an early design decision to facilitate extensibility by the community where developers can just register adapters that interface with systems such as scientific resources and that may be hosted within RSMS but also by other parties (i.e., there is no need for plugging in code).

Building such an infrastructure presents several interesting issues and challenges not only from a practical but also from a conceptual point of view. These issues and their implications (that will be addressed on this work) can be summarized as follows:

- **Lack of common conceptual model for scientific resources.** Given the vast number of resources in the "space" of resources, it becomes difficult for services to handle each resource-specific operations (e.g., searching, publishing, changing access rights) and properties (metadata). The lack of a common conceptual model for resources makes services to be limited to specific resources, i.e., the ones hard-coded in the service implementation. It also means that clients need to integrate the different data sources and give the semantics to the resources and their relations. What this implies is that a common conceptual model should be general enough to cope with the potential requirements of the services on top, and simple enough to be useful. Defining such a conceptual model with a proper compromise is part of this research work.
- **Heterogeneous interfaces.** Scientific resources can be provided by "traditional" or "novel" data sources. Interfacing with such heterogeneous data sources normally requires clients to implement the access for each of them, given that the interfaces of these sources have different signature details. In some cases, sources are not even exposed via APIs or meant to be crawled (for example, getting citations from

Google Scholar), then requiring clients to implement wrappers and raising some other problems like limited access, banning, etc. Therefore, to facilitate building services on top, the platform should abstract the specifics of the different data sources and provide a common interface.

- **Difficulty to extend sources available.** Extending the sources available to the services implies in most cases changing the service implementation to introduce the required support (e.g., adding a new citations source to a service that computes citation-based indexes). To avoid this problem, the platform should incorporate an extensibility model that allows us to extend the sources available without introducing changes in the platform.
- **Maintenance cost and scalability.** Associated to the above problems is the maintenance cost. Adding new sources, maintaining wrappers as they may become obsolete, providing support to new resources; all of them imply effort. Moreover, as the number of services grows, the scalability of the platform becomes a problem. Therefore, to sustain a platform like the one we intend, it is necessary to reduce the effects on costs and performance. A preliminary model based on distribution of effort and computation is presented in this paper.

In the following we outline a research plan towards the design and development of models and systems for RSMSs.

2 Use case: Liquid Journals

As part of this research work, we will develop a use case based on a new model for scientific knowledge dissemination: *liquid journals* (LJ) [2]. This use case will allow us validate the concepts, models and system developed in the context of this work.

In a nutshell, LJ is a system capable of querying the Web for “interesting” and “relevant” scientific contributions, and that can evolve to include new contributions as they appear on the Web (or become “relevant” to the journal). In the context of LJ, scientific contributions can range from traditional peer-reviewed journals, to blogs and scientific datasets. It is up to the editor to specify the type and qualities of contributions he/she wants in the liquid journal.

Thus, the effort in developing the liquid journals will be on the definition of a query language capable of capturing the notions of “interestingness” and “relevance”, and on the development of the underlying query engine on top of the RSMS, capable of merging results from various data sources (e.g. search engines, social bookmarking services, …), filtering and grouping the results according to the query definition and to rank them according to their relevance. The RSMS will provide seamless access to the scientific data sources and a conceptual model for scientific contributions.

3 Research Space Management Systems

In order to overcome the issues we have presented before, we build the platform around the abstraction of scientific resource and provide a general and extensible model: the *resource space*. Then, from the infrastructure point of view, we provide a collaborative-extensible and distributed platform. An overview of the preliminary model and platform is presented in the following.

3.1 The Resource Space

RSMS is based on the notion of viewing every possible kind of scientific contribution available on the web as a scientific resource. Under this assumption, the web is a (scientific) resource space and the RSMS manages – and simplifies – access to these resources.

A *resource* can be any artifact we can refer to by an URI and that is accessible over the Web (e.g., documents, experiments, but also metadata from citeUlike and Google Scholar, etc). These resources are managed by potentially different service providers (e.g., Google Docs, Google Scholar, ...). We refer to these service providers as *resource managers*. Then, the third element we consider is the *action*. Actions describe the services provided by resource managers and that allow us to operate with the resources (e.g., to share or search documents, or more complex actions such as crawling a web site for scientific metadata). On top of this we provide set of abstractions, to free upper layers of implementing resource specific operations.

Incidentally, these abstractions are natural extensions of the basic elements. Thus, the first abstraction we consider is the *resource type*, which characterizes families of resources with similar behavior. Analogously, *resource manager types* denote general classifications of resource managers, such as archives, search engines, control version systems, etc. Then, the *action type* provides a common interface for semantically equivalent actions. For example, to "change access rights" in both Wiki and GoogleDocs regardless the differences in their "signature" detail.

On top of these constructs, we define entities and specific metadata and operations for scientific resources that correspond to common resources and actions that services need to perform. In this preliminary model, we consider the following scientific entities: scientific contributions, people, communities, events. The mapping of those entities to the resource space is performed by defining particular resource types that encapsulate the properties, relations and behavior of those. It is possible, however, to extend these entities following our extensibility model below.

3.2 Extensibility model

In general terms, the approach we follow is to provide a set of core modules that can manage the *adapters* and access to resource managers through these adapters. Adapters are provided by third parties and made available to the upper layers trough the registration service of the RSMS. This allows us to extend the sources available without introducing changes into the platform, so making the platform easier to maintain.

The RSMS extensibility approach, the resource manager and the concept of resource type collectively support a flexible binding approach that can range from static to dynamic binding to both adapters and (for services using the RSMS) to resources. Static binding to adapters is implemented by restricting (for a given or all RSMS clients) access to a given (set of) resources to go through a specified adapter - and therefore using a specific mapping between generic actions and actual operations.

However in general it is possible to change dynamically the adapter we use to access a given resource: the mappings are specified and the adapters are registered, this is transparent to RSMS clients. Besides load balancing, the key benefit here is reliability and the ability to leverage the community to maintain a complex distributed system. For example, the RSMS could switch to another adapter in case the one in use becomes obsolete. Note that dynamic binding here is "provider-enabled" in that the provider of the adapter makes sure to define the mapping with the resource type actions.

4 State of the Art

With the goal of providing access to scientific resources available on the web, search engine technology has been explored and applied to scientific content [3]. Examples of those are Google Scholar and Citeseer, which provide only a partial view of what we consider as scientific contributions. Besides efforts on standardizing the access to scientific content (e.g., the OAI-PMH protocol, www.openarchives.org/pmh), these search engines, as well as most of scientific services, provide heterogeneous interfaces and metadata format. Moreover, in the scope of this research we also consider the problem of operating on data sources (not only search).

Thus, a relevant area is that of service integration and interoperability, where research on service compatibility [4], and recently on models and frameworks for service integration, replaceability and interoperability has been done (e.g., [5, 6]). These works serve as foundation over which we build our strategy for providing uniform interface for operating on scientific resources at different levels of abstraction. In this work, however, we take the *resource* perspective, paying particular attention on modeling the scientific services associated to the resources, while ensuring universality, maintainability and scalability.

In other dimension we have the information integration, which is important as most relevant scientific services imply getting metadata from sources with different metadata schema. In this area, important research has been done on techniques for data integration, organized in two general approaches: *global-as-view* [7] and *local-as-view* [8]. From these research works, we have learnt about strategies that can be combined and applied to our proposed model of resource space, at different levels: from a resource manager to it correspondent resource manager type, and from this to our uniform model of scientific resources.

Finally, a highly relevant work to RSMS is that of *Dataspaces*, which extend DBMS concepts to reach heterogeneous data sources by allowing integration to be done in an incremental fashion [9]. In particular, building applications over this layer allows searching and operating with multiple data sources using a common interface. As this framework is general, therefore, it does not provide any particular modeling for scientific resources. In addition, there is not focus on extensibility, on providing resource type abstractions, which hides the specificities of the resources and allows operating on resource of the same type with the same set of operations.

5 Conclusion and Research Plan

In this paper we have introduced a conceptual framework for RSMS. This system is inspired by the idea of having a homogeneous view of a space of resources, in which those resources can be provided by different and heterogeneous resource managers on the Internet. In this context, we have also introduced a preliminary model for the resource space applied to scientific entities. The innovative aspects of the proposed abstraction layer rely on a combination of *universality*, which allows us to manage any web-accessible resource; *accessibility*, in terms of homogeneous and source-independent access to resources; *simplicity*, in terms of the general model and of the abstractions used, and *extensibility*, which is a property of both the model (which allows us to define different new resources and actions at different levels of abstractions) and of the architecture (that allows us to plug new resource managers).

The current status of this research work is a preliminary conceptual model of the resource space, and architecture and working prototype of the RSMS core (based on the resource management module of Gelee [10]).

As future work, we will define a detailed conceptual model of the scientific entities and integrate it with the resource space conceptual model. We will extend the RSMS core to cope with advanced features such as dynamic adapter binding and distributed adapters and, once finished, we will bootstrap the platform with a set of common research services that we will derive from the liquid journals use case. The platform described in the context of this research will constitute the foundation of the technical architecture of the Liquidpub project.

References

1. Casati, F., Giunchiglia, F., and Marchese, M. Liquid Publications: Scientific Publications Meet the Web. http://eprints.biblio.unitn.it/archive/00001313/01/073.pdf
2. Baez, M., Casati, F. Liquid Journals: Knowledge Dissemination in the Web Era. http://project.liquidpub.org/liquid-documents
3. Lossau, N. Search Engine Technology and Digital Libraries, Libraries Need to Discover the Academic Internet. D-Lib Magazine, 2004, Vol. 10, No. 6, ISSN 1082-9873.
4. B. Benatallah, F. Casati, and F. Toumani. *Web services conversation modeling: A Cornerstone for EBusiness Automation.* IEEE Internet Computing, 8(1), 2004.
5. S. R. Ponnekanti and A. Fox. *Interoperability among Independently Evolving Web Services.* Middleware '04. Toronto, Canada, 2004.
6. B. Benatallah, F. Casati, D. Grigori, H. R. M. Nezhad, and F. Toumani. Developing adapters for web services integration. In Procs of CAiSE 2005.
7. J. D. Ullman. Information integration using logical views. In Proc. of ICDT'97, volume 1186 of Lecture Notes in Computer Science, pages 19–40. Springer, 1997
8. Kirk, T., Levy, A. Y., Sagiv, Y., Srivastava, D.: The Information Manifold. In: Proc. of AAAI 1995 Spring Symp. on Information Gathering (1995) 85–91
9. Franklin, M., Halevy, A., and Maier, D. 2005. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.* 34, 4 (Dec. 2005), 27-33.
10. Baez, M., Casati, F., and Marchese, M. Universal Resource Lifecycle Management. ICDE/WISS 2009, Shanghai, China.

Online Testing of Service-Oriented Architectures to detect State-based Faults

Michaela Greiler
supervised by Hans-Gerhard Gross and Arie van Deursen

Delft University of Technology, Delft, The Netherlands,
{m.s.greiler|h.g.gross|arie.vanDeursen}@tudelft.nl

Abstract. Service-oriented architectures have found their way into industry to enable better business-to-business cooperations. With this software architecture new challenges for software development and testing appeared.
In this proposal we discuss the problem of testing these complex, and distributed systems in dedicated test environments. We argue that state and configuration of the production system can influence system behavior in an unexpected way, and that test environments do not reflect the final system adequately. Therefore, we propose the development of an online testing method to detect state-based faults, and discuss related research challenges and solutions.

1 Introduction

A service-oriented architecture (SOA) is a software architecture that aims at reusability and interoperability by exposing functionality as loosely coupled services, that can be composed to form business processes. Testing such systems is aggravated by stakeholder separation: third party services are black-boxes for integration testers, and influence in their evolution and maintenance is not permitted. Dynamic features, like ultra-late binding or dynamic composition of services, allow a flexible and dynamic way of composing a concrete system just at runtime, but restrict the ability to test a priori. The context in which a service will be used is often unknown at the service's development time, imposing problems for service testers to predict and foresee possible requirements and usage scenarios. The limited testability of service-oriented systems imposes the need of reconsidering and redesigning traditional, and inventing new testing methods and processes, as mentioned in [5, 8, 10, 12].

This proposal suggests online testing as a method to identify state-based faults in online reconfiguration of service-oriented systems. Online testing is testing carried out in the realm of the production system, and in parallel to its operational performance. In our context, state comprises information about previous executions, the concrete set of installed and active programs, as well as history of preceding configuration activities. Reconfiguration includes evolution of business processes, services, business logic and changes in the execution environments.

The proposal is structured as follows: first, related work is summarized. Section 3 states the problem and formulates the research hypothesis. The related research challenges and possible solutions are discussed subsequently. The research plan is outlined in Section 5, followed by the evaluation plan in Section 6. The expected outcomes are summarized in Section 7.

2 Related Work

Influenced by the need of new testing methods and techniques for SOA, many approaches on unit, integration, interoperability, and regression testing can be found in the literature [4, 6]. We are focusing especially on related work concerning integration and online testing, where in particular the lack of control of integrated services e.g., to access the service code or to put a service in test mode, and the lack of information of integrated services e.g., to generate stubs, aggravate testing.

Similar to online testing is in vivo testing [7], also dubbed as perpetual testing[1], used to assess live applications after deployment. However, while in vivo testing focuses on performing unit tests, our proposal focuses on integration testing of service compositions. The scope of the test naturally affects the test approach, the test cases required, and their number. Online testing, as we see it, does not replace offline integration testing, but offers an efficient approach to address reconfiguration faults that cannot be identified in the offline test environment.

Bertolino et al. [1, 2] have also looked at the field of online testing, with interoperability testing, approaching the testability problem by redirecting service invocations to stubs. They suggest that the originally passive registry should take over an active audition role, and check the correctness of a service against its specification before listing it. Services already active in the SOA do not take part in the testing process. In contrast, our method will check the service implementation against expectations (specification) of requesting services, and not against its own specification.

Heckel et al. [11] discuss an approach, in which a central third party, called the discovery agency, checks that a service conforms with its specification, i.e., a model, by performing full behavioral and functional tests, before it is registered. Later on, other services can compare their requirements against the model provided by a discovered service. Their assumption is that the service could be faulty, either unintentionally because of insufficient testing, or intentionally for introducing malicious services. This differs from our online testing in which we focus on integration faults that are not easily caught using just offline testing. Further, it is not obvious in their description how a service checks its requirements model against the service specification model.

To run a test in the production system, undesired side effects need to be avoided, thus the system must provide test isolation, which has been addressed by [3, 9, 13]. Our approach to achieve test isolation differs from these approaches where a test sensitive component is cloned, replaced with a proxy, either in test mode or operational mode, or where testing is completely aborted. In our online testing framework, a component is not cloned, but two versions of a service are instantiated, whereby inheritance is used to allow test isolation, which makes the solution less expensive. Since both services are available at the same time, testing does not block the operational service, and can be executed concurrently.

All testing approaches abstract away from the complex and heterogeneous environments in which the services are operated, and do not mention their influence on systems' behavior.

[1] http://www.ics.uci.edu/∼djr/edcs/PerpTest.html

3 Problem Statement and Research Hypothesis

Testing of SOA systems means testing of organization-spanning systems-of-systems. In most cases it is not feasible to set up an accurate test environment, that represents the final environments of all involved parties in which services are used. Many parts of third parties have to be stubbed or mocked, even if there is a lack in provided information. Different and heterogeneous execution environments and their configuration have to be recreated, whereas a clean initial system state still does not reflect the production environment. Beside resource limitations, also time restrictions hinder adequate set-ups, leading to the risk that even tested software fails during online integration in the production environment.

We want to determine how the state of a system can influence its behavior in an unexpected way, and develop an online testing method for identifying state-based faults during online reconfiguration of service-oriented architectures.

On account of this the research hypothesis is: "Online testing is an effective strategy for detecting state-based faults in service-oriented systems."

4 Research Challenges and Proposed Solutions

An efficient online testing method to detect reconfiguration faults has to address following challenges.

What are the typical reconfiguration faults in SOAs, and which state and configuration produce those, even if software and services have been offline tested?

The main challenge is to determine which state and configuration data influence program execution in a SOA environment. This has to be narrowed down by the question: "Which information is often left out in the set-up of test environments?".

To define possible faults it has to be clear how reconfigurations of service-centric systems are propagated to local and remote systems. Possible faults have to be partitioned in those that can easily be detected during offline testing, and those that cannot. Latter faults have to be subdivided again, based on the question: "Which errors are already handled by the middleware, and which have to be handled by services and systems themselves?"

Regarding state and configuration, security and authentication policies are often not known or accurately implemented in a test environment. Many errors, like service unavailability and deployment errors, will be detected and partly solved by the middleware. During composition, and service execution, typical integration faults, like interface, data format, and communication protocol mismatches are to be expected. Further, dependency errors caused by evolution of parts of the system can occur, especially if some of the services and systems involved are stubbed or mocked in the test environment. It is important to analyze the impact of software evolution on the existing systems, either local or remote, and which layers of a SOA are affected. This can reach from changes in the business process layer, to the business logic layer and also to changes in the execution environment. Also application tolerance, which is the tolerance of multiple applications for each other, has to be addressed if a new services is integrated. Changes in the backend of a service, be it the business logic or the configuration of the execution environment, can have rippling effects on local or even remote systems. These changes can be for instance, policies, new installed software, changes of database schema or changes of the database management system. Such modifications can cause service level agreement violations, because the network load increases, the new database management system responses slower, or access is now denied.

How should an effective online testing framework for SOA integration and system testing be designed, and what are the requirements for such a framework?

Challenging during design and development of an online testing method, are test isolation, performance, and application enhancement effort. If a system's behavior is assessed online, it has to be guaranteed that testing is isolated from the production system, and no unintended side effects appear. Test execution can only take place if no performance decreases or even unavailability of service are experienced. Enhancement costs for applying online testability mechanisms have to correspond to the positive impact accompanied with them. This implies that the tradeoff of online testing is corporeal. "But how can the effectiveness of an online testing method be assessed and measured?" Evaluation could be based on the capability of online testing to reveal faults in contrast to offline testing.

Information regarding the state of execution environments and infrastructures (e.g. application server, messaging bus, etc.) should support online testing to reveal faults. "But what is an accessible and uniformed way to provide state information to testing entities?"

Monitoring capabilities can be used to notify about reconfiguration changes in the system. An observer service, installed on each server involved in the SOA environment, could provide access to this information to interested and authenticated testing entities.

How can test cases, oracles, and system models be derived or generated?

In many SOA environments, abstract business models, business process models or UML documents, as well as service descriptions are present. Those represent a good foundation to derive system models and test cases. Test cases should not represent full behavioral tests, but address aspects that could not been tested offline.

To solve the oracle problem, functional equivalence testing can be applied. The running system is considered to be correct and the test output for testing the new, adapted version of the system is compared with the output of the old, stable system. Capture and replay techniques can help to increase observability, and controllability of the testing process, because old input and output sequences are recorded and reused ad libitum.

5 Research Plan

We have implemented an online testing framework providing test isolation in a case study, based on OSGi (Open Service Gateway Initiative), to evaluate the fault finding capabilities of our online testing approach. We could gain a first insight in state-based faults, mainly caused by missing required packages, wrong class bindings, and inconsistencies in the overall system. Services have been consistent with their specification, but when integrated in the execution environment, inconsistencies between different services emerged.

In the coming year, we plan to set up a SOA laboratory in order to examine configuration and state information that is significant during system reconfiguration, and to understand better how these influence the testing effectiveness and accuracy. Based on the results we want to develop a fault taxonomy for faults caused by evolutions in distributed, heterogeneous runtime environments.

In the next year, our online testing method has to be enhanced to identify these faults, and to be applicable in industrial SOA environments (e.g., IBM product suite).

Subsequently, we will explore capabilities to reduce the application enhancement effort. This will include automatic generation of test cases, and of testability artifacts allowing runtime service assessment. With this outcome, the online testing method will be extended to an open source online testing framework, providing support for test generation and execution. Throughout the research, the outcomes will be evaluated based on case studies.

6 Evaluation Plan

The SOA laboratory is a network of distributed servers, on which services, installed in different execution environments, form actual applications, inspired by industry. Communication is handled by a centralized service registry and bus.

This laboratory will be used to conduct representative case studies, following Yin [14], to evaluate our online testing method. Update scenarios of different parts of the system will function to determine the accuracy of the test environment to prevent failures, and the fault finding capabilities of our method.

In collaboration with our industrial partner, we also want to assess the effort to prepare an application, with support of our framework, for online testing. For that, we will measure, for example, the required time and the number of necessary changes in the existing system. A comparison of effort and fault finding capabilities should indicate if online testing has a positive tradeoff.

7 Expected Outcomes

The resulting contributions of the doctoral studies include:

1. A new approach for online testing to detect state-based faults.
2. A series of fault models classifying faults that appear especially in the production environments.
3. Empirical evidence concerning the effectiveness of the proposed approach.
4. An open source tool for online test support, including generation and execution of test cases.

Our results should corroborate the hypothesis by indicating the effectiveness of online testing to detect state-based faults during system reconfiguration.

References

1. A. Bertolino, L. Frantzen, A. Polini, and J. Tretmans. Audition of web services for testing conformance to open specified protocols. In *Architecting Systems with Trustworthy Components*, pages 1–25. Springer, 2006.
2. Antonia Bertolino, Guglielmo Angelis, Lars Frantzen, and Andrea Polini. The plastic framework and tools for testing service-oriented applications. In *Software Engineering: International Summer Schools, ISSSE 2006-2008, Salerno, Italy, Revised Tutorial Lectures*, pages 106 – 139, Berlin, Heidelberg, 2009. Springer-Verlag.
3. Daniel Brenner, Colin Atkinson, Oliver Hummel, and Dietmar Stoll. Strategies for the runtime testing of third party web services. In *SOCA '07: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications*, pages 114–121, Washington, DC, USA, 2007. IEEE Computer Society.
4. A. Bucchiarone, H. Melgratti, S. Gnesi, and R. Bruni. Testing service composition. In *Proceedings of the 8th Argentine Symposium on Software Engineering (ASSE'07) Mar del Plata, Argentina*, pages 29–31, August 2007.
5. Gerardo Canfora and Massimiliano Di Penta. Testing services and service-centric systems: Challenges and opportunities. *IT Professional*, 8(2):10–17, 2006.
6. Gerardo Canfora and Massimiliano Penta. Service-oriented architectures testing: A survey. In *Software Engineering: International Summer Schools, ISSSE 2006-2008, Salerno, Italy, Revised Tutorial Lectures*, pages 78–105, Berlin, Heidelberg, 2009. Springer-Verlag.
7. Matt Chu, Christian Murphy, and Gail Kaiser. Distributed in vivo testing of software applications. In *ICST '08: Proceedings of the International Conference on Software Testing, Verification, and Validation*, pages 509–512, Washington, DC, USA, 2008. IEEE Computer Society.
8. Schahram Dustdar and Stephan Haslinger. Testing of Service-Oriented Architectures - A practical approach. In *Object-Oriented and Internet-Based Technologies*, pages 97–109. Springer Berlin/ Heidelberg, 2004.
9. Alberto González, Éric Piel, and Hans-Gerhard Gross. Architecture support for runtime integration and verification of component-based systems of systems. In *1st International Workshop on Automated Engineering of Autonomous and run-time evolving Systems (ARAMIS 2008)*, pages 41–48, L'Aquila, Italy, September 2008. IEEE Computer Society.
10. Michaela Greiler, Hans-Gerhard Gross, and Khalid Adam Nasr. Runtime integration and testing for highly dynamic service oriented ICT solutions – an industry challenges report. In *TAIC-PART '09: Proceedings of the Testing: Academic & Industrial Conference on Practice And Research Techniques*, pages 51–55, Windsor, UK, 2009. IEEE.
11. Reiko Heckel and Leonardo Mariani. Automatic conformance testing of web services. In *Proc. Fundamental Approaches to Software Engineering (FASE)*, pages 34–48. Springer, 2005.
12. Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. Service-oriented computing: State of the art and research challenges. *Computer*, 40(11):38–45, 2007.
13. Dima Suliman, Barbara Paech, Lars Borner, Colin Atkinson, Daniel Brenner, Matthias Merdes, and Rainer Malaka. The morabit approach to runtime component testing. In *COMPSAC '06: Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, pages 171–176, Washington, DC, USA, 2006. IEEE Computer Society.
14. R. K. Yin. *Case Study Research, Design and Methods*. Sage Publications, Beverly Hills, CA, second edition, 1994.

Ensuring Cost-Optimal SLA Conformance for Composite Service Providers

Philipp Leitner

Supervised by: Schahram Dustdar

Distributed Systems Group
Vienna University of Technology
Argentinierstrasse 8/184-1
A-1040, Vienna, Austria
lastname@infosys.tuwien.ac.at

Abstract. For providers of composite services, service level agreements (SLAs) provide a means to guarantee a certain service quality to prospective customers. Usually, violating SLAs is associated with costs. However, the means necessary to ensure SLA conformance also generate costs. Oftentimes, it is therefore optimal from a business perspective to violate certain SLAs sometimes, instead of trying for the high (and expensive) road of always satisfying each one. In this paper we will sketch a framework for the prediction of SLA violations and for determining whether an adaptation of the process makes sense economically. If this is the case adaptation actions are triggered, which adapt the composition on either on instance, structural or environmental level. The ultimate goal is to implement a closed-loop system, which self-optimizes the costs resulting from SLA violations.

1 Introduction

Service-oriented architectures are at their core about the integration of systems. This new paradigm is used by Software-as-a-Service providers, which deliver basic IT functionality such as customer relations management or business intelligence as composite services. One important notion for the seamless integration of such outsourced IT services are agreements about the quality that these services need to provide (QoS), typically defined within legally binding Service Level Agreements (SLAs). SLAs contain Service Level Objectives (SLOs), concrete numerical QoS objectives which the service needs to fulfill. If SLOs are violated, agreed upon consequences (usually taking the form of penalty payments) go into effect. However, fulfilling SLAs can also lead to costs for the service provider (e.g., because the composite service provider needs to use more expensive services itself, or because of the costs inherent to optimizing its service composition). It is therefore not trivial for the provider to decide to what extend the service's SLAs should be fulfilled, or which SLAs should (temporarily) be violated for economical reasons. Even more, these decisions should optimally be automated, to allow for fast reactions to changes in the business environment.

In this overview paper we will present a high-level framework for optimizing adaptations of service compositions with regards to SLA violations. We use techniques from the area of machine learning to construct models allowing the system to predict SLA violations at runtime and decide which adaptation actions may be used to improve overall performance. Adaptation can happen on instance level (for one instance only), on structural level (for all future instances), or on environmental level (e.g., migrating the composition engine to a machine with better hardware). An optimizer component decides if applying these changes makes sense economically (i.e., whether the costs of violating the SLAs are bigger than the adaptation costs). If this is the case the respective actions are applied in an automated way. At its core, this system is a closed-loop self-optimizing system [1], with the target of minimizing the total costs of adaptations and SLA violations for the service provider.

The work described in this paper is currently ongoing. However, some important fundamental work has already been published. In [2] earlier work regarding the monitoring of QoS of Web services is presented. Our work on VRESCo [3] forms the basis for the proposal presented here, providing core services such as support for dynamic rebinding. In [4] we have presented first results regarding the identification of factors influence of business process performance, which is related to the generation of prediction models, which has been extended in [5]. The remaining PhD research will be led by two key research questions: (1) How can the factors that influence the performance of a composition be identified, modeled and analyzed, in order to enable prediction of SLA violations at runtime, and trigger adaptations to prevent these violations? (2) How can the tradeoff between preventing violations and the costs of doing so be best formalized, especially considering that many adaptation actions may be interleaved and combined? Even though first steps exist (see Section 3), to the best of our knowledge, these questions have not been answered sufficiently in literature so far. We plan to validate the outcomes of the thesis using a case study, by showcasing how self-optimization can prevent SLA violations both short- and long-term, and comparing the total costs for the service provider with and without the proposed system (e.g., using a standard WS-BPEL process). We hope to show that the service provider will be able to both reduce the number of SLA violations using automated adaptation (even if not all violations are prevented) as well as to reduce the total costs of adaptations and violations.

The remainder of the paper will be structured as follows. Section 2 contains the main contribution of the paper, a description of a system for cost-optimal adaptation of composite services. In Section 3 we give a brief overview over relevant related work. Finally, Section 4 will conclude the paper.

2 Approach Overview

A high-level overview of our approach is depicted in Figure 1. The system implements an optimization cycle in the Autonomic Computing [1] sense, i.e., it follows the basic steps *Monitor* (monitoring the service composition, i.e., measuring QoS values and process instance data), *Analyze* (generating prediction models), *Plan* (evaluating based on the generated models, the available adaptation actions and the current SLAs of the provider if there are possible optimizations for the composition), and *Execute* (applying these optimizations). The managed element is the *Service Composition*, while four other components (*Composition Monitor*, *Composition Analyzer*, *Cost-Based Optimizer* and *Adaptation Executor*) implement the autonomic manager.

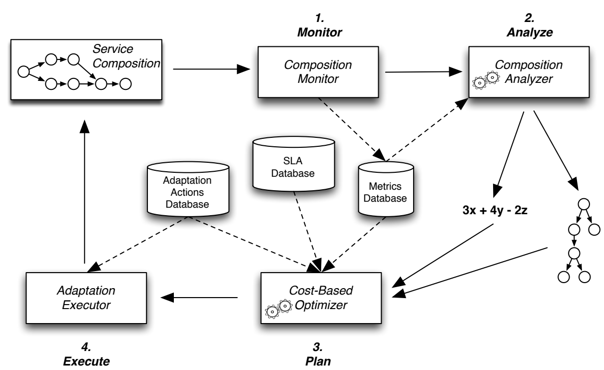


Fig. 1: A Self-Optimizing System for Cost-Optimal SLA Conformance

Our approach is based on the following assumptions: (1) service providers have explicit SLA(s) with their customers, including concrete numerical target values for SLOs and penalty payments for SLA violations; these payments can be staged, i.e., more severe SLA violations can lead to more severe penalties, and (2) there is a database of possible adaptations available, including the costs of these adaptations; costs can be both one-time costs (such as a downtime) or continually (such as increased costs for using more expensive services). The implementation of this database of possible adaptation actions needs to be supported by strong tools, which allow for the generation of the most important actions in a semi-automated way. Costs of adaptation actions can be derived using simulation or analysis of historical data. In the following, we are assuming an autonomous system, however, in some situations the role of the optimizer or executor can also be adopted by a human.

Composition Monitor: The foundation of all work presented here is the ability to gather accurate runtime data. This includes: (1) QoS metrics such as response time or availability of the services used, (2) runtime payload data, such as customer identifiers or ordered items, and (3) technical parameters of the execution environment, such as the availability of the composition engine, or the CPU load of the machine running the composition. The *Composition Monitor* component is used to collect this monitoring data from various sources (e.g., an external

QoS monitor [2] or technologies such as Windows Performance Counters[1]), consolidate it and store it to a metrics database.

Composition Analyzer: The data collected by the *Composition Monitor* is then used to generate prediction models for SLA violations. Prediction models are used to estimate at runtime if a given running instance is going to violate one or more of its SLAs. They are associated with checkpoints in the composition model, at which the prediction is done. Simply put, a prediction model is a function which uses all execution data which is already available at the checkpoint and, if possible, estimations for all missing data, and produces a numerical estimation for every target value in the provider's SLAs as output. We present first results regarding this component in [5].

Cost-Based Optimizer: The *Cost-Based Optimizer* can be seen as the core of the system. This component needs access to all SLAs, as well as a database of possible adaptation actions. The optimizer has to fulfill two important tasks in the system. Firstly, it uses the prediction models generated before to predict concrete QoS values for every running instance and compares the predicted values with the respective SLOs. If SLA violations would occur it checks the Adaptation Actions database for any possible action to prevent the violation, and applies them if it is cost-efficient to do so. This involves solving an optimization problem to decide which combination of actions both prevents most SLA violations and is cheapest to implement. Secondly, if more than a certain threshold of SLA violations (in a given time frame) have been monitored, the component tries to improve the composition itself, i.e., it tries to optimize the composition for every future instance. The main difference is that on composition level more possible adaptation actions exist (mainly because adaptation on this level is less time-critical, so that adaptations which involve e.g., a system downtime are also feasible).

Adaptation Executor: The *Adaptation Executor* is responsible for applying the adaptation actions as planned by the *Cost-Based Optimizer*. Generally, we consider the classes of adaptation actions (action classes) depicted in Figure 2. As discussed before, adaptation can happen either on instance (level 1, i.e., adaptations which affect only a single instance) or structural level (level 2, i.e., adaptations which affect all future instances), and can consist of rebinding base services (R*, i.e., switching from one used service to another), restructuring the composition (S*, e.g., parallelizing some parts of the composition) or adapting the execution environment (E, e.g., upgrading the virtual machine running service composition). Generally, actions of type E always affect all future instances, and are therefore only applicable on level 2.

Obviously, the concrete execution of adaptation actions depends greatly on the action class. For applying adaptations of type R* we can use the means provided by the execution environment (such as dynamic rebinding as discussed in

[1] `http://msdn.microsoft.com/en-us/library/aa373083(VS.85).aspx`

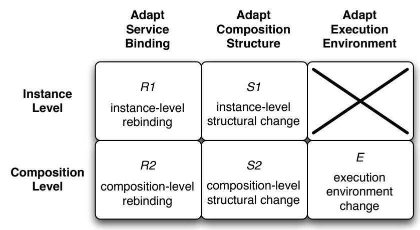


Fig. 2: Classes of Adaptation Actions

earlier work [3]). For S* more complex means are necessary, for instance adaptation techniques such as BPEL'n'Aspects [6]. Currently, changes of class E are mostly executed manually. However, for some changes of this class automation has been made possible by the recent rise of Cloud Computing. If, for instance, the execution environment is hosted in the Amazon Elastic Compute Cloud[2] it is possible to automatically adapt (some) parameters of the hosting environment via the Amazon S2 API.

3 Related Work

We will now briefly discuss some key related work. QoS monitoring of atomic services is discussed in [2, 7]. Our *Composition Monitor* will partially be based on these results. Monitoring of composition instance data has been discussed in [8]. However, these works do not explicitly cover SLA monitoring, which is the scope of [9]. The authors use an event-based approach to monitor QoS, which is in line with the ideas we have used in [4]. This work also pioneers the idea of SLA impact analysis, which is related to the tasks that our *Composition Analyzer* has to fulfill. Another basic building block of this component is the work presented in [10], which discusses the prediction of QoS (again using an event-based model). Self-adaptation of compositions, another core topic in our work, is discussed in [11]. The MASC system presented there adapts itself in order to recover from failures and improve reliability, however, this system does not try to predict problems and prevent them proactively. Optimizing service compositions with regards to overall QoS is an often-discussed topic, with some seminal work dating back to 2004 [12]. In contrast to this approaches, in our system optimization is done with regard to specific SLOs, which are currently violated, and taking into account the tradeoff between the costs of SLA violations and the costs of adaptation.

4 Conclusions

In this paper we have sketched the architecture of a closed-loop system, which autonomously optimizes service compositions with regard to SLA violations,

[2] `http://aws.amazon.com/ec2/`

taking into account the costs caused by the adaptation. Our next steps will be the implementation of an first end-to-end system, which includes prototypes for all four main components of the system.

Acknowledgements

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

References

1. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. IEEE Computer **36**(1) (2003) 41–50
2. Rosenberg, F., Platzer, C., Dustdar, S.: Bootstrapping Performance and Dependability Attributes of Web Services. In: ICWS '06: Proceedings of the IEEE International Conference on Web Services. (2006) 205–212
3. Michlmayr, A., Rosenberg, F., Leitner, P., Dustdar, S.: End-to-End Support for QoS-Aware Service Selection, Invocation and Mediation in VRESCo. Technical report, TUV-1841-2009-03, Vienna University of Technology (2009)
4. Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Leymann, F., Dustdar, S.: Monitoring and Analyzing Influential Factors of Business Process Performance. In: EDOC'09: Proceedings of the 13th IEEE International Enterprise Distributed Object Computing Conference. (2009)
5. Leitner, P., Wetzstein, B., Rosenberg, F., Michlmayr, A., Dustdar, S., Leymann, F.: Runtime Prediction of Service Level Agreement Violations for Composite Services. In: NFPSLAM-SOC'09: The 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing. (2009)
6. Karastoyanova, D., Leymann, F.: BPEL'n'Aspects: Adapting Service Orchestration Logic. In: ICWS 2009: Proceedings of 7th International Conference on Web Services. (2009)
7. Moser, O., Rosenberg, F., Dustdar, S.: Non-Intrusive Monitoring and Service Adaptation for WS-BPEL. In: Proceedings of the 17th International Conference on World Wide Web (WWW'08). (2008) 815–824
8. Wetzstein, B., Strauch, S., Leymann, F.: Measuring Performance Metrics of WS-BPEL Service Compositions. In: ICNS'09: Proceedings of the Fifth International Conference on Networking and Services, IEEE Computer Society (April 2009)
9. Bodenstaff, L., Wombacher, A., Reichert, M., Jaeger, M.C.: Monitoring Dependencies for SLAs: The MoDe4SLA Approach. In: SCC '08: Proceedings of the 2008 IEEE International Conference on Services Computing. (2008) 21–29
10. Zeng, L., Lingenfelder, C., Lei, H., Chang, H.: Event-Driven Quality of Service Prediction. In: ICSOC '08: Proceedings of the 6th International Conference on Service-Oriented Computing. (2008) 147–161
11. Erradi, A., Maheshwari, P., Tosic, V.: Policy-Driven Middleware for Self-Adaptation of Web Services Compositions. In: Middleware'06: Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware. (2006) 62–80
12. Zeng, L., Benatallah, B., H.H. Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering **30**(5) (2004) 311–327

Remote Sensing Service Chain Self-Evolution Method

Haifeng Li
Supervised by Qing Zhu
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University
430079 Wuhan, China
hfli6135@lmars.whu.edu.cn

Abstract. In order to facilitate continuous and quick adaptation to the change of environment and use requirements, this paper presents a self-evolution method for the remote sensing information service chain to keep effectiveness and robustness of service chain with less modification, and to maximize all users' utilization under concurrent user requirements. According to the driver forces of service chain change, we partition the self-evolution method into three levels: (1) a semantic distance based relatedness analysis and min-conflict heuristic based service chain reconfiguration method to adapt user requirement change; (2) a sensitivity analysis and robust optimization based method to keep maximum stability of service chain in dynamic environment; (3) a non-cooperative game approach for multi-service chain cooperation optimization under tasks concurrent.

1 Introduction

As satellites of Earth Observing System (EOS) currently beam down several hundreds of gigabytes data daily, the inconsistency between powerful data instruments and incompetent data process becomes ever more standing, which is making this field "data-rich but analysis-poor" [1]. The key reason leads to it is not lack of applications to handle with these data, but mechanisms how to aggregate the applications which are distributed in Internet extensively (hence be looked as remote sensing services) together and cooperate them to satisfy the need of the data analysis. This is so-called remote sensing service chain[2] through service composition. Compared with generic Web service composition, remote sensing services have some typical features as follows:

Complex in user requirements, for examples: real-time monitor for forestry fire, coast, and flood; concurrent in user requirements, for instance, in Sichuan Wenchuan Earthquake, we must evaluate earthquake damage and monitor coast, landslides, and barrier lakes at the same time.

Rich in data dimensions. Data in remote sense with dimensions of spatiality, temporal, image resolution, sensor type, and image spectrum, makes it harder to be described and more complicated in processing flow. What's more, mass remote sensing images make the service chain more sensitive to response time.

Complex in remote sensing processing. Remote sensing service composition has been constrained by more strict process semantic; Computation-intensive feather in remote sensing also makes remote sensing processing more time consumed.

So, remote sensing service chain should be flexible enough to effectively adapt to fast change of use requirements and environment, through frequently refining their structure. The existent methods to generate services have some disadvantages are as following:

Lack of mechanism to adapt user requirements change via local reconfigure at function level which are known as abstract services[3]. The state-of-art service composition approaches[4] are facing more and more serious bottlenecks of effectiveness and stability, since new service chain must be generated from “scratch” for each requirement. Those methods are also known as “first principle”. Distinguish with it, another way is how to make use of relativity between remote sense service chains and reuse knowledge about similar, already solved problems successfully. These methods are always known as “second principle”[5], which aim to make service chain generate more effectively and execute more stably. Although there are some researches generate service chain by case based reasoning[6-7], they do not take into account strict process constraints in remote sensing. What’s more, how to measure similarity between cases accurately and refine service chain effectively are still open questions.

Lack of robust adapting to dynamic environment at capability level known as concrete services[3] which usually modeled as QoS constraint based optimization. The service chains are much more sensitive to services and transportation network performance, because data-intensiveness and computing-intensiveness are essential features in remote sensing. A small perturbation in QoS dimension of services and transport network will make former optimization solution become infeasible. There are many researches dynamic modify service chain through runtime monitor and re-planning technology[8]. But, because of high dynamic and uncertain of services and transport network in nature, the dynamic modification may be too frequent, and leads to unstableness and decrease of performance of service chain execution. Therefore, we still are short of quantization model to estimate the influence of QoS perturbation on service chain. The mechanism how to keep service chain robust in dynamic and uncertain environment is unclear.

Lack of optimal mechanisms to deal with concurrent user requirements. The existing optimal composition approaches search optimization solution[9] under QoS constraints (such as response time, cost, stability and available) via “selfish” way. Yet, these methods only take single used requirements into account, not adapt to applications like remote sensing emergency and disaster response where concurrent task happened frequently. Concurrent tasks competing optimal services leads to conflict problem and decreased performance of all service chains, which are known as “tragedy of the commons”. A key problem is how to reduce the conflict caused by concurrency tasks to make all service chains reach the optimization at the same time.

In conclusion, in the face of high dynamic environment, dynamic and concurrent of user requirements, the challenge is: how service chains adapt to user requirements and environment to keep effective and robust of service chain with lesser modification and how to implement multiple service chains cooperation optimization under concurrent tasks to maximize all users’ utilization. Hence, we put forward a novel method to solve them from self-evolution viewpoint.

2 Remote Sensing Service Chain Self-Evolution Method

The basic conception behind remote sensing service chain self-evolution method is: it is a self-adaptive behavior responds to exterior dynamics factors, through frequent revise structure, function and capability of service chain, with completeness, minimization and consistency.

Exterior factors dynamics refer to user requirements, services runtime environment which include temporal disability of service, modification of services QoS and network QoS, and so on.

Completeness, refers to if it is feasible to change from current service chain to others, then, we always can find the post-evolution service chain.

Minimization, refers to achieve the evolution process with minimum service chain changing. The minimization has two means here: maximum reuse existent service chain and least revision that establishes the upper and lower limits of the sensitivity interval and finds a robust solution with less sensitivity to dynamic environment.

Consistency, includes function consistency and capability consistency, i.e. evolution process must satisfied constraints such as function constraints and QoS constraints.

2.2 Service chain Self-Evolution Method

We reduce self-evolution method to two levels and three hierarchies according to the driving forces mentioned above, shown as table 1. The abstract service chain made up of abstract service to satisfied user function requirement. Concrete service chain made up of concrete service to which mapping or binding very abstract service. The proposed research methods consist of three aspects as follows, also shown as fig. 2.

Table 1. The basic idea of service chain evolution.

Service chain level	Question	Basic idea	Hierarchy
Single service chain	How to adapt user requirement changing	Choosing a most similarity service chain by user requirements relatedness analysis, and fast reconfiguring by local revising based on reuse knowledge about similarity, already solved problems successfully.	Abstract service chain
	How to keep service chain robust in dynamic and uncertain environment	Analysis the influence of QoS perturbation including QoS preference, QoS constraints, and services QoS on service chain; set up a robust optimization model to keep service composition optimal solution more stability.	Concrete service chain
Multi-service chain	How to make all service chain reach optimization at the same time in concurrent situation.	Modeling competition relationship between tasks by non-cooperative game, which assures maximizing all tasks' utilization under multi-task conflict condition.	

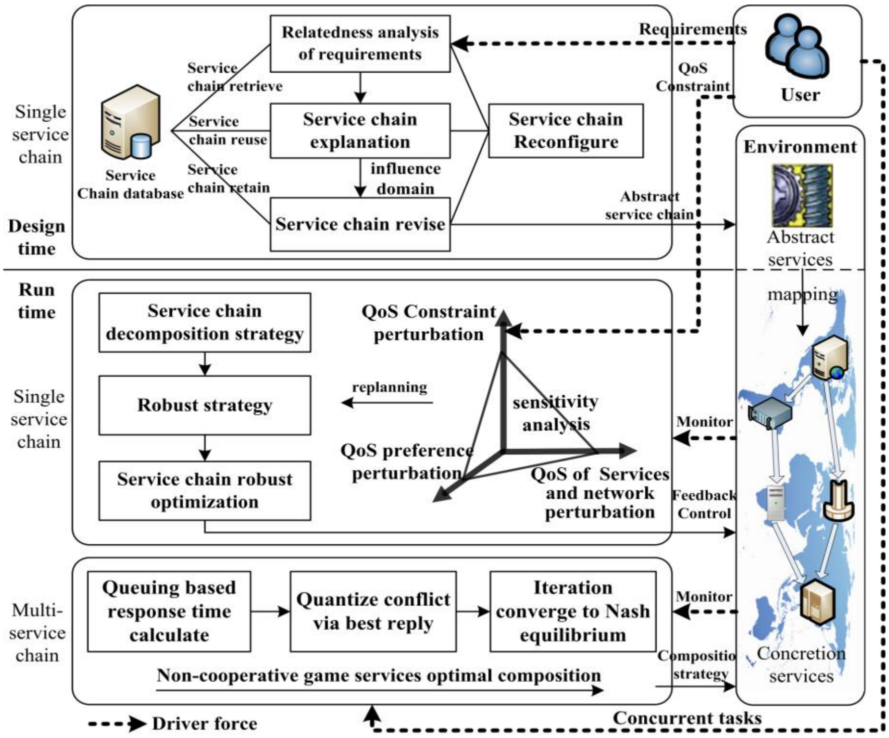


Fig. 1. Architecture of service chain self-evolution method

(1) Adapt to user requirement

A pair $\langle SC, R \rangle$ describe the service chain (the former), and corresponding user requirement (the latter). We put forward a reconfiguration approach to adapt user requirement change. The basic idea is find a most similarity service chain by analyze relativity between current user requirement, say R^n and former one, say R^0, and then get a new service chain through minimal local modify.

We model user requirement as and/or graph, in which each node describes term ontology in the domain. Relativity analysis of user requirements hence reduces to the graph similarity question, and divides it to two main steps:

The first step is estimate the similarity between each node in the graph. The term ontology describe is-a, and is-kind relationship about terms. We use term semantic distance by network distance models to implement it[10]. The second step is estimate the similarity by Hausdorff distance on graph level.

After relatedness analyses, we accomplish service chain reconfiguration by min-conflict heuristic based regression algorithm to search a minimal influence domain solution, and prove be A*. Regression search algorithms start from changed requirements to initialization input by user, and min-conflict heuristic guide search process to promise search efficiency away from "combination explore" problem, which is first applied to solve of constraint satisfying and schedule problem by local revising [11].

The reconfiguration is adding and deleting services in essence. We define unsatisfied requirements and preconditions (according to IOPE) of services as sub tasks. Adding a new service to current services chain will reduce the number of sub tasks, but introduce some new sub tasks-the preconditions. Hence, min-conflict is designed to maximize former, and minimize the latter.

(2) Robust optimization in dynamic environment

We quantitative analysis the influence of QoS perturbation on service chain performance via (mix) integer linear programming model shown as(1), where f is object functions, g is inequality constraints and h equality constraints. They are linear functions. The service QoS model used here is Zeng's five dimensions model: execution price, execution duration, reputation, successful execution rate, availability[9]. Based on this, we cast QoS preference, QoS constraints, and QoS of services and network to profit parameters, left-hand side and left-hand side of constraints, respectively, and establish the upper and lower limits of the sensitivity interval through sensitivity analysis. This is the process decide if the re-planning should be triggered.

$$\begin{aligned} \max_{x \in \Re^n} \quad & f^T(v), \\ s.t. \quad & g_i(v) \geqslant a, i \in \mathrm{I}, \\ & h_j(v) = b, j \in \varepsilon, \end{aligned} \tag{1}$$

Then, we set up robust optimization model via minimax criterion[12] to deal with the uncertain of service QoS. According to service chain execution state, the service chain can be classified three states: executed, executing, and waiting for execute. The key principle is only the waiting for execute state services will be considered. We structure QoS value uncertainly by intervals defined by lower and upper bounds, and minimax criterion defined as finding among all globe optimization solution the one that minimizes the maximum from them. Minimax criterion based robust optimization approach can give a conservation solutions based on the anticipation that the worse-case will happened.

By strategy motioned above, we decrease service chain sensitivity to uncertain environment and reduce re-planning frequency by robust optimization solution.

(3) Concurrent tasks optimization

The key idea to solve the problem in concurrent tasks optimization is finding an optimal composition method which insures the service chain composition strategy of each task is the best even considering other tasks' strategy. Hence, a non-cooperative game based mathematics model is proposed to analysis competition relationship between tasks through best reply function. In this game, every user as actor, the optimization combination solution as strategy, and user object function f as utility.

To achieve this game, best reply based iteration algorithm is proposal. Best reply is used to quantize conflict between tasks to assure each task finds optimal composition strategy adapting to other tasks'. It includes two parts: "the best" and "the reply". We modify global model of Zeng[9] by apply a new queuing theory based response time compute approach to calculate optimization strategy. And, we defined remainder computer capability to describe "the reply". Iteration algorithm is the process where the users update their optimization strategy one after the other until the predefined threshold reached. It will converge to Nash equilibrium, which maximizing all task's utilization under multi-task conflict condition.

The non-cooperative game based service optimization composition method is making sure every task has the highest service utility under the other user's strategy considered, and we anticipate the game converge to Nash equilibrium, in which no task could get a better utility by changing its service composition strategy selfish.

3 Conclusion

Remote sensing service chain self-evolution method is a self-adaptive behavior to exterior factors dynamics, through frequent revise structure, function and capability of its. We have done some experiments on user requirements change and tasks concurrent scenarios and our methods show good performance in former and good convergence and better practice utility of all tasks in latter.

Next step, we will focus on our sensitivity analysis and robust optimization based method to test capability of keeping service chain stability in dynamic environment.

References

1. Clery, D. and Voss, D.: All for one and one for all. Science, 308(6):757, (2005)
2. Alameh, N.: Chaining geographic information Web services. IEEE Internet Computing, 7(5):22-29, (2003)
3. Canfora, G., Penta, M.D., Esposito, R. and Villani, M.L.: A framework for QoS-aware binding and re-binding of composite web services J. Syst. Softw., 81(10):1754-1769, (2008)
4. Sirin, E., Parsia, B., Wu, D., Hendler, J. and Nau, D.: HTN Planning for Web Service Composition Using SHOP2. In Proc. Proceedings of 2nd International Semantic Web Conference(ISWC'03). pp. 20-23. Springer, Sanibel Island, Florida, USA (2003)
5. Koehler, J.: Planning from second principles. Artificial Intelligence, 87(1-2):145-186, (1996)
6. Klusch, M. and Renner, K.-U.: Fast Dynamic Re-Planning of Composite OWL-S Services. In Proc. Proceedings of the 4th IEEE European Conference on Web Services (ECOWS 2006) pp. 134-137. IEEE CS Press, Zurich, Switzerland (2006)
7. Recio-García, J.A., Diaz-Agudo, B. and González-Calero, P.A.: A Distributed CBR Framework through Semantic Web Services. In: Bramer, M., Coenen, F.andAllen, T. (eds.): In Proc. Proceedings of the Twenty-fith SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, AI 2005. pp. 88-101. Springer (2005)
8. Canfora, G., Penta, M.D., Esposito, R. and Villani, M.L.: QoS-Aware Replanning of Composite Web Services. In Proc. Proceedings of the IEEE International Conference on Web Services (ICWS'05). pp. 121-129. IEEE Computer Society (2005)
9. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J. and Chang, H.: QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, 30(5):311-327, (2004)
10. Cross, V.: Fuzzy Semantic Distance Measures Between Ontological Concepts. In Proc. Processing IEEE Annual Meeting of the Fuzzy Information 2004 NAFIPS '04. pp. 635-640 (2004)
11. Sosic, R. and Gu, J.: Efficient Local Search with Conflict Minimization: A Case Study of the n-Queens Problem. IEEE Trans. Knowl. Data Eng., 6(5):661-668, (1994)
12. Beyer, H.-G. and Sendhoff, B.: Robust optimization – A comprehensive survey. Comput. Meth. Appl. Mech. Eng., 196(33-34):3190-3218, (2007)

Remotely Sensed Image Processing Service Automatic Composition

Xiaoxia Yang
Supervised by Qing Zhu
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University
430079 Wuhan, China
yangxx2003@126.com

Abstract: Aiming at the correctness, quality and efficiency of remotely sensed image processing service composition for geospatial applications, a remotely sensed image processing service composition approach is proposed. It includes three main algorithm: (1) remotely sensed image processing service chaining based on heuristic search to composite services into a meaningful order; (2) knowledge navigated remotely sensed image processing service classification and selection, which using data mining to select an appropriate service for specific user requirement; (3) remotely sensed image processing service selection with response time to meet the response time requirement from user.

1 Introduction

As the world changes more rapidly, the demand for up-to-date information for resource management, environment monitoring, urban planning, crisis management and emergency response are increasing exponentially. Remote sensing technology has been widely recognized for contributing to geospatial information efforts. As remote sensing technologies become ever more powerful with multi-platform and multi-sensor, hundreds of terabytes of image data is available daily. But in many cases, raw remotely sensed images are not directly useful without further processing. There are more and more needs to aggregate remotely sensed image processing to satisfy the increasing demands of various applications. Because the remotely sensed image processing demands large-scale, collaborative processing and massive storage capabilities, the effect and efficiency of the remotely sensed image processing is far from the user's expectation. More intensive and more complex tasks make us "data-rich but analysis-poor"[1].

The emergence of Service Oriented Architecture (SOA) may make this challenge manageable. The SOA allows cooperation of data and process components among different organizational units and supports reusability and interoperability of components through the network to satisfy more and more complex applications. Remotely sensed image processing services are modular components of remote sensing applications that are self-contained, self-describing and can be published, located and invoked across a network to access and process remote sensing data from a variety of sources[2].

The service composition on demand has become a hot topic. It is urged to encapsulate all processing function into services and recombine them with service chain. Service composition, the process of creating the service chain through composing a collection of services, is required. The various requirements of users can be achieved by combining different existing data and services into a value-added service chain. Automatic service composition, if successful, can be of great value to the geospatial user community.

Remotely sensed image processing problems usually involve large and heterogeneous data and multiple computation steps and service providers. The composition approaches are illustrated by an example from the domain of remote sensing based change detection. Figure 1 illustrates the process of change detection with remotely sensed image, which generally consists of such steps as image acquirement, pre-processing, image registration, and change detection.

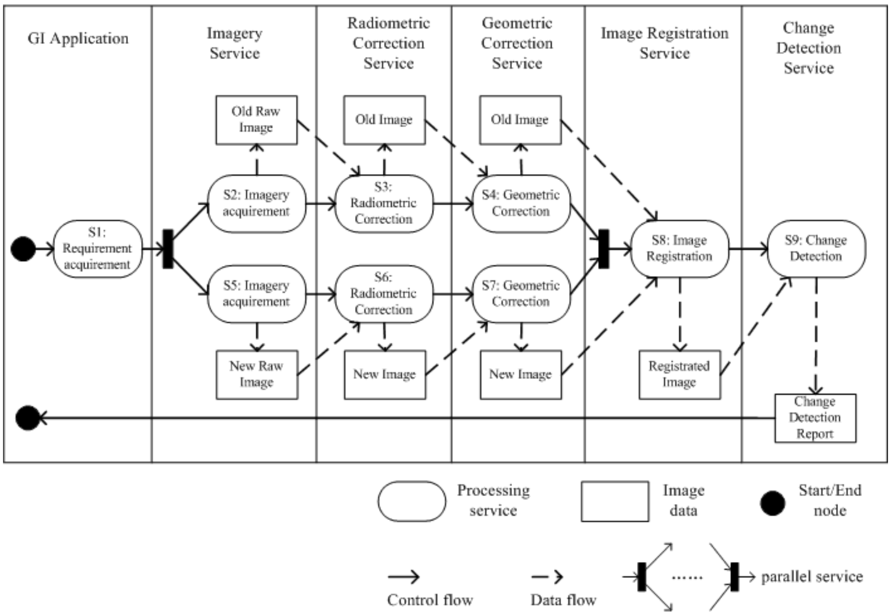


Fig. 1 Change detection service chain

2 Problem

The key to achieve automation relies mainly on solutions to three issues[3]:

(1) How to make remotely sensed image processing services interoperable both syntactically and semantically;

(2) How to automatically select, based on the syntactic and semantic descriptions, the most appropriate data and services;

(3) How to assemble them to build the service chain.

Various users require composition of different processing services in a meaningful order to solve specific problem. A key challenge in promoting widespread use of remotely sensed image processing services in the geospatial applications is to automate the construction of a chain or process flow that involves multiple services and highly diversified and distributed data.

For remotely sensed image processing service chaining, an important problem is finding suitable services and to select the most suitable one according the task requirements. Because of the lack of the adequate knowledge of remotely sensed image service selection, most existing keyword-based and ontology-based service selection approaches are not very effective and efficient.

Quality of service(QoS) is an important factor that should be considered in the process of service composition. In general, there would be many individual processing services offering similar functionality but with different qualities. In current QoS-based service research, the response time is commonly considered as a certain value [4]. As to remotely sensed image processing services, it is difficult to estimate and manage response time for two additional reasons: in most remotely sensed image applications such as in case of emergency, the requirement for response time of the service chain is very rigorous; to allow for a dynamic network environment and the uncertainty of processing service QoS, response time will vary within a range rather than being a specific value.

3 Proposed solution

Aiming at the correctness, quality and efficiency of processing service composition for remote sensing application, a remotely sensed image processing service composition approach is proposed in the dissertation.

Table 1. The basic idea of service composition

Service level	Question	Basic idea
Abstract service	How to assemble them to build the service chain	Remotely sensed image processing service chaining based on heuristic search
	How to automatically select the most appropriate services	Knowledge navigated remotely sensed image processing service classification and selection
Concrete service		Remotely sensed image processing service selection with response time

Remotely sensed image processing service chaining based on heuristic search construct service chain through two steps: 1) dynamically constructs a complete service dependency graph for user requirement on-line; 2) AO* based heuristic searches for optimal valid path in service dependency graph. These services within the service dependency graph are considered relevant to the specific request, instead of overall registered services. The second step, heuristic search is a promising approach for automated planning. Starting with the initial state, AO* uses a heuristic function to select states until the user requirement is reached.

It is a major challenge to select appropriate remotely sensed image processing services to satisfy both functional and non-functional requirement. In an attempt to facilitate and streamline the process of service selection, selecting services involves two main selection processes, abstract service selection and concrete service selection. Abstract service selection pick out a service to perform certain remotely sensed image processing function. Concrete service selection choose from among a set of functionally equivalent ones for each abstract service.

At the abstract service level, a novel service selection approach, knowledge navigated remotely sensed image processing service classification and selection, is proposed in the dissertation. It consists of three main steps: service cluster pre-process, knowledge discovery and service selection. In the pre-process, the similar candidate services are grouped into clusters called service clusters as the objects to be selected to distinguish services detailed and sharp down the search space. In the second step, decision–tree, a classification method, is used to discover the latent relation between specific task and services. Finally, the knowledge is used to decide which service clusters to perform the task. By learning from domain experts' interactions and other analysts' experiences with various services, the approach will help analysts determine what service set most satisfies the immediate problem at hand.

At the concrete service level, based on the probability theory, we constructs the probability response time estimation model to construct the constraint between the expected value, the variance, and the user's requirement of response time. By using the critical path method (CPM), some critical services which have direct and crucial effects on the response time guarantee are picked out. To satisfy the constraint, a service selection algorithm of service selection is proposed to reselect appropriate remotely sensed image processing services. Thus, the optimized service chain can meet the response time requirement of the user with higher probability than before.

4 Conclusion

Our work includes two main contributions. The first is a remotely sensed image processing service chaining algorithm. The second contribution is a service selection algorithm on two levels. Three algorithms are described in this work: (1) remotely sensed image processing service chaining based on heuristic search to composite services into a meaningful order; (2) knowledge navigated remotely sensed image processing service classification and selection, which using data mining to select an appropriate service for specific user requirement; (3) remotely sensed image processing service selection with response time to meet the response time requirement from user.

References

1. Clery, D. and D. Voss, All for one and one for all. Science. 308(6): p. 757, 2005
2. Onchaga, R. Modelling for quality of services in distributed geoprocessing. in 20th Congress of the ISPRS. 2004
3. Di, L., A framework for developing Web-service-based intelligent geospatial knowledge systems. Journal of Geographic Information Sciences. 11(1): p. 24-28. 2005
4. Zeng, L.Z., et al., QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering, 30(5): p. 311-327. 2004.