Hamid Reza Motahari-Nezhad
Farouk Toumani Yannis Velegrakis (Eds.)

ICSOC PhD Symposium 2008

Co-located with 6th International Conference on Service Oriented Computing (ICSOC)

Sydney, Australia, 1 December 2008

Proceedings

Sponsor: IBM Research, USA

Preface

Service oriented computing (SOC) has rapidly transformed from a vision, in the beginning of the century, to realisation in paradigms such as Web services, Software-as-a-Service (SaaS) and cloud services. While this has provided the industry and practitioners with the opportunities for a new generation of products and services, it has brought forward a tremendous amount of challenges and open issues for researchers. The International Conferences on Service Oriented Computing (ICSOC) is a pioneering event for researchers, practitioners and industry leaders to discuss and share the success and achievements in this area.

The ICSOC PhD Symposium, as part of the ICSOC conference, is an international forum for PhD students working in the broad areas of service computing, web services and service engineering to present and discuss emerging research problems and ideas on how to tackle these issues. The forum is intended to bring together PhD students and give them the opportunity to present and discuss their research in a constructively critical atmosphere. The symposium operates in a workshop format, giving PhD students an opportunity to showcase their research and providing them with feedback from senior international researchers and peer PhD students. The goals of the ICSOC PhD Symposium event are

- To provide fruitful feedback and advice to the selected Ph.D. students on their research thesis.
- To provide the opportunity to meet experts from different backgrounds working on topics related to service oriented computing field.
- To interact with other PhD students and stimulate an exchange of ideas and suggestions among participants.

The ICSOC PhD Symposium 2008 is attended by prominent researchers in the field of service oriented computing who will actively participate in and contribute to the reviews and discussions. The symposium in Sydney is the 4th PhD Symposium of a series held in conjunction with the ICSOC conferences in Vienna, Austria (2007), Chicago, USA (2006), Amsterdam, The Netherlands (2005). This year we have received 24 submissions from 9 countries and 3 continents: 14 from Australia-Pacific, 8 from Europe, 1 from Asia and 1 from Africa. Each paper has been reviewed by three members of program committee. The submissions were evaluated based on five criteria: originality (novelty), problem significance, technical/scientific quality, and presentation/style. Each reviewer also provided a detailed review intended for improving the research work of the students. Finally, the program chairs selected 12 papers for the presentation in the symposium based on the recommendations of the program committee members. The selected papers cover a wide range of topics in SOC from service engineering, quality of service, security and trust to service modeling and composition.

We would like to express our appreciation to the authors for their submissions and the program committee members for their hard work. We would also like to thank IBM Research, USA that has supported the symposium by generously covering part of the registration,

travel and accomodation expenses of a number of students attending the symbosium. We specifically thank Nirmal Mukhi from IBM Research for helping to streamline the this process.

November 2008

Hamid Reza Motahari Nezhad,
Farouk Toumani, Yannis Velegrakis (PC Chairs)

Programme Chairs

Hamid Reza Motahari Nezhad
Farouk Toumani
Yannis Velegrakis

Programme Committee

Karim Baina
Claudio Bartolini
Florian Daniel
Marlon Dumas
Schahram Dustdar
Aditya Ghose
Andreas Hanemann
Seyyed Vahid Hashemian
Chengfei Liu
Heiko Ludwig
Annapaola Marconi
Michael Papazoglou
Julien Ponge
Regis Saint-Paul
Michael Sheng
Halvard Skogsrud
Kenneth Wang
Eric Wohlstadter
Andreas Wombacher
Xiaohui Zhao

External Reviewers

Lianne Bodenstaff
Volha Kerhet
Evan Morrison
Heorhi Raik
Adina Sirbu
Jian Yu

Contents

Advice to Service-Oriented Computing Research Students (Invited Talk)

Marlon Dumas

University of Tartu, Estonia and Queensland University of Technology, Australia
marlon . dumas @ ut.ee

Extended Abstract

Service-Oriented Computing (SOC) is a study field at the crossroads of several other fields, such as software engineering, information systems, databases, distributed computing and Internet computing. It is also a rather dynamic field since it is constantly being reshaped by ongoing industrial developments. As a result, SOC is by no means an easy field to undertake a PhD project. This talk will discuss a number of factors that PhD students in the field need to be mindful of when scoping, planning and executing their project.

One such factor is the importance of balancing the scientific requirement of rigor with the aim of being relevant and impacting the practice of SOC. This latter goal can be pursued by undertaking an *applicability check* [1] early on during the PhD project in order to assess the importance and timeliness of the problem and expected outcomes. This effort should be complemented with an ongoing reflection on the accessibility and suitability of the proposed solution, without neglecting the need to follow research methods that withstand scientific scrutiny.

Another factor to be considered is the importance of being validation-aware throughout the PhD project, rather than postponing all the validation effort until the end. This goal can be achieved in a two-pronged manner: (i) by preparing the validatory work and collecting data right from the start of a PhD project; and (ii) by alternating design and development steps with validation steps. Different validation methods are commonly used in the field, including prototype implementation, case study, performance experiment, mathematical proof, user experiment, survey. Each of them has its own trade-offs and scope of applicability [2]. Factors worth consideration when selecting a validation method is the type of research question, the type of research outcome, and more broadly, where the PhD project stands in the spectrum between design and empirical science.

Finally, when selecting and scoping a PhD project, one has to be mindful of current agendas and trends in the field. These agendas and trends will be briefly discussed during the talk.

References

1. M. Rosemann and I. Vessey. Toward Improving the Relevance of Information Systems Research to Practitioners: The Role of Applicability Checks. *MIS Quaterly* 32(1):1–22, March 2008.
2. M Shaw. What Makes Good Software Engineering Research. *International Journal of Software Tools for Technology Transfer*, 4(1):1–7, 2002.

Distributed and Scalable QoS Optimization for Dynamic Web Service Composition

Mohammad Alrifai

L3S Research Center
Leibniz University of Hannover, Germany
alrifai@L3S.de

Supervised by: Prof. Dr. tech. Wolfgang Nejdl
L3S Research Center
Leibniz University of Hannover, Germany
nejdl@L3S.de

Abstract. Web service composition requests are usually combined with end-to-end QoS requirements, which are specified in terms of non-functional properties (e.g. response time, throughput and price). The goal of QoS-aware service composition is to select the best combination of services that meet these end-to-end requirements, while maximizing the value of a pre-defined utility function. This problem can be modeled as a multi-dimension multi-choice 0-1 knapsack problem, which is known as NP-hard in the strong sense. Existing solutions that rely on general purpose solvers suffer from poor performance, which render them inappropriate for applications with dynamic and real-time requirements. Moreover, global optimization techniques assume a centralized system model, which contradicts with the distributed and loosely-coupled environment of web services. The aim of this thesis is to develop scalable QoS optimization solutions that fit better to the distributed environment of web services. The idea is to decompose global constraints into local constraints that have to be fulfilled by a set of distributed service brokers. A solution that combines global optimization and local selection techniques is proposed.

1 Introduction

The service-oriented computing paradigm and its realization through standardized Web service technologies provide a promising solution for the seamless integration of business applications to create new value-added services. Industrial practice witnesses a growing interest in this ad-hoc service composition. With the growing number of alternative web services that provide the same functionality but differ in quality parameters, the composition problem becomes a decision problem on the selection of component services with regards to functional and non-functional requirements. In this work, we look at the non-functional requirements, namely quality of service parameters in composing web services.

1.1 Motivating Scenario

Consider for example the personalized multimedia delivery scenario in Figure 1. A PDA user requests the latest news from a service provider. Available multimedia content includes a news ticker and topical videos in MPEG 2 only. The following services are are required to serve the user's request: a transcoding service for the multimedia content to fit the target format, a compression service to adapt the content to the wireless link, a text translation service for the ticker, and also a merging service to integrate the ticker with the video stream. The user request can be associated with some end-to-end QoS requirements (like bandwidth, latency and price). The service composer has to ensure that the aggregated QoS values of the selected services match the user requirements. Dynamic changes due to changes in the QoS requirements (e.g. the user switched to a network with lower bandwidth) or failure of some services (e.g. some of the selected services become unavailable) can occur at run-time. Therefore, a quick response to adaptation requests is important in such applications.

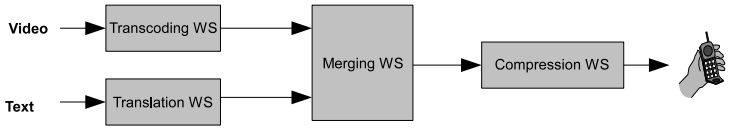

Fig. 1. Composition of Multimedia Web Services

1.2 Local vs. Global QoS Optimization

Two general approaches exists for the QoS-driven service composition: *local* optimization and *global* optimization. In the local optimization approach, one service is selected from each service class independently based on its local utility value. This approach is very efficient as the time complexity of the local optimization approach is linear with respect to the number of service candidates. However, local optimization cannot satisfy end-to-end QoS requirements (like maximum total response time). On the other hand, the global optimization approach aims at solving the problem on the composite service level. This approach seeks the service composition, which maximizes the overall utility value, while guaranteeing global constraints. The global optimization problem can be modeled as *Multi-Choice Multidimensional Knapsack problem* (MMKP), which is known to be NP-hard in the strong sense [1]. Therefore it can be expected that any exact algorithm that solves MMKP has an exponential effort [2], which is not suitable for real time applications. Moreover, global optimization approaches rely on centralized computation, which is not feasible for the distributed and dynamic environment of web services.

1.3 Expected Contribution

The aim of this PhD thesis is to address the performance and scalability issues of QoS aware service selection by applying a scalable distributed heuristic that

combines global and local optimization techniques. The approach contributes the following results to the state of the art:

1. Decomposition of global QoS constraints into local constraints is modeled as a mixed integer program. The size of the resulting program is independent on the number of service candidates and hence can be solved more efficiently than existing MIP-based solutions.
2. Selection of component services is solved using guided local optimization. The local optimization is performed for each service class independently and in parallel to further improve the performance.
3. Extensive evaluation of the approach by comparing its performance and quality with existing exact and heuristic solutions. The evaluation will be done by simulation as well as in a large real world environment, e.g. PlanetLab.

2 Related Work

The QoS-based web service selection and composition in service-oriented applications has recently gained the attention of many researchers [3–6]. In [4] the authors propose an extensible QoS computation model that supports open and fair management of QoS data. The work of Zeng at al. [3] focuses on dynamic and quality-driven selection of services. The authors use global planning to find the best service components for the composition. They use linear programming techniques to find the optimal selection of component services. Similar to this approach Ardagna et al. [5] extend the linear programming model to include local constraints. Linear programming methods are very effective when the size of the problem is small. However, these methods suffer from poor scalability due to the exponential time complexity of the applied search algorithms [7, 2]. In [6] the authors propose heuristic algorithms to find a near-to-optimal solution more efficiently than exact solutions. The time complexity of the heuristic algorithm for the combinatorial model is polynomial, and exponential for the graph model. Despite the significant improvement of these algorithms compared to exact solutions, both algorithms do not scale with respect to the number of web services and remain out of the real-time requirements.

3 A Distributed Approach for Web Service QoS Optimization

We divide the QoS-aware service composition problem into two sub-problems that can be solved more efficiently in two subsequent phases. In the first phase, we use global optimization techniques to find the best decomposition of global QoS constraints into local constraints on the component service level. In the second phase, we use local selection to find the best component services that satisfy the local constraints from the first phase.

We assume an architecture consisting of a *service composer* and a number of *service brokers*. The service composer instantiates a composite service in collaboration with the service brokers. Each service broker is responsible for managing

QoS information of a set of web service classes. A list of available web services is maintained by the service broker along with registered measurements of their non-functional properties, i.e. QoS attributes, like response time, throughput, price etc. For the sake of simplicity we assume in this paper that each service class is maintained by one service broker. The two phases of our approach are described in the next subsections in more details.

3.1 Decomposition of Global QoS Constraints

In order to avoid discarding any service candidates that might be part of a feasible composition, the decomposition algorithm needs to ensure that the local constraints are relaxed as much as possible while meeting global constraints. To solve this problem, we divide the quality range of each QoS attribute into a set of discrete quality values, which we call "*quality levels*". We then use mixed integer programming (MIP) to find the best combination of these quality levels for using them as local constraints. The size of our MIP model is much smaller than the size of the MIP model in [3, 5] as the number of decision variables in our case is much smaller than the number of variables in their model. Therefore, our MIP model can be solved much faster.

Quality Levels: In this paper, we use a simple method for constructing the quality levels. For each service class S_i, we divide the quality range of each of the m QoS attributes into d quality levels: $q_{ik}^1, ..., q_{ik}^d$, $1 \leq k \leq m$ as follows:

$$q_{ik}^z = \begin{cases} Qmin(i,k) & \text{if } z = 1 \\ q_{ik}^{z-1} + \frac{Qmax(i,k) - Qmin(i,k)}{d} & \text{if } 1 < z < d \\ Qmax(i,k) & \text{if } z = d \end{cases} \quad (1)$$

where $Qmin(i,k)$ and $Qmax(i,k)$ are the local minimum and maximum values, respectively, for the kth attribute of the service class S_j. We then assign each quality level q_{ik}^z a value between 0 and 1, which indicates the probability p_{ik}^z that using this quality level as a local constraint would lead to finding a solution. The probability p_{ik}^z for the zth level of q_k at S_i is computed as follows:

$$p_{ik}^z = h/l \quad (2)$$

where h is the number of service candidates satisfying q_{ik}^z and l is the total number of service candidates at S_i.

MIP Formulation: The goal of our MIP model is to find the best decomposition of QoS constraints into local constraints. Therefore, we use a binary decision variable x_{ik}^z for each local quality level q_{ik}^z such that $x_{ik}^z = 1$ if q_{ik}^z is selected as a local constraint for the QoS attribute q_k at the service class S_i, and $x_{ik}^z = 0$ otherwise. To ensure that only one quality level is selected from the set

of d levels of the QoS attribute q_k at the service class S_i, we add the following set of constraints to the model:

$$\sum_{z=1}^{d} x_{ik}^{z} = 1, \forall i, \forall k, 1 \leq i \leq n, 1 \leq k \leq m$$

where n is the number of service classes and m is the number of QoS constraints. Note that the total number of variables in the model equals to $n * m * d$, i.e. independent on the number of service candidates per class l. By ensuring that the number of quality levels d is small enough such that $m * d \leq l$ we can ensure that the size of our MIP model is smaller than the size of the model used in [3, 5]. The selection of the local constraints must ensure that global constraints are still satisfied. Therefore, we add the following set of constraints to the model:

$$\sum_{i=1}^{n} \sum_{z=1}^{d} q_{ik}^{z} * x_{ik}^{z} \leq c'_k, \forall k, 1 \leq k \leq m$$

The objective function of our MIP model is to maximize the probability that the selected local constraints will lead to finding a feasible composition. Therefore, using (2) the objective function can be expressed as follows:

$$\text{maximize} \prod_{i=1}^{n} \prod_{k=1}^{m} p_{ik}^{z} \ , 1 \leq z \leq d \tag{3}$$

We use the logarithmic function to linearize (3) in order to be able to use it in the MIP model:

$$\text{maximize} \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{z=1}^{d} ln(p_{ik}^{z}) * x_{ik}^{z} \tag{4}$$

By solving this model using any MIP solver methods, we get a set of local quality levels that we use in the second phase for guiding local search.

3.2 Local Search

We use the local constraints obtain from the first phase as upper bounds for the QoS values of component services. Web services that violate these local constraints are skipped from the search. We then sort the qualified services by their utility values and select the top service from each class. In order to evaluate the multi-dimensional quality of a given web service we use a Multiple Attribute Decision Making approach: i.e. the *Simple Additive Weighting (SAW)* technique [8] to compute the utility value of the service. The utility computation involves scaling the values of QoS attributes to allow a uniform measurement of the multi-dimensional service qualities independent of their units and ranges. We compute the distance between the quality value $q_k(s_{ji})$ of a given service candidate s_{ji} and the maximum value $Qmax(j,k)$ in its class S_j and compare

it with the distance between the maximum and minimum overall quality values that can be obtained by any composition: $Qmax'(k) = \sum_{j=1}^{n} Qmax(j,k)$, $Qmin'(k) = \sum_{j=1}^{n} Qmin(j,k)$. This scaling method ensures that the evaluation of service candidates is globally valid, which is important for guiding local search in order to avoid local optimums. The scaling process is then followed by a weighting process for representing user priorities and preferences.

We compute the utility $U(s_{ji})$ of the i-th service candidate in class S_j as:

$$U(s_{ji}) = \sum_{k=1}^{r} \frac{Qmax(j,k) - q_k(s_{ji})}{Qmax'(k) - Qmin'(k)} \cdot w_k \tag{5}$$

with $w_k \in \mathbb{R}_0^+$ and $\sum_{k=1}^{r} w_k = 1$ being the weight of q_k to represent user's priorities.

4 Conclusion and Future Work

This PhD thesis is aimed at the development of distributed and scalable solutions to the global QoS optimization problem for web service compositions. Unlike existing approaches that model the problem as a conventional combinatorial optimization problem, we model the problem as a distributed optimization problem by exploiting the special characteristics and structure of the web service environment. Current results of this work indicate a very promising improvement over existing solutions. The next steps of this work include extending the existing model to support different styles of web service compositions and QoS constraints. We also plan to evaluate the performance of our approach against existing exact and approximate solutions by extensive simulations as well as in a large real world environment, e.g. PlanetLab.

References

1. Pisinger, D.: Algorithms for Knapsack Problems. PhD thesis, University of Copenhagen, Dept. of Computer Science (1995)
2. Parra-Hernandez, R., Dimopoulos, N.J.: A new heuristic for solving the multichoice multidimensional knapsack problem. IEEE Transactions on Systems, Man, and Cybernetics, Part A **35**(5) (2005) 708–717
3. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: WWW. (2003) 411–421
4. Liu, Y., Ngu, A.H.H., Zeng, L.: Qos computation and policing in dynamic web service selection. In: WWW. (2004) 66–73
5. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. IEEE Trans. Software Eng. **33**(6) (2007) 369–384
6. Yu, T., Zhang, Y., Lin, K.J.: Efficient algorithms for web services selection with end-to-end qos constraints. TWEB **1**(1) (2007)
7. Maros, I.: Computational Techniques of the Simplex Method. Springer (2003)
8. Yoon, K..P., Hwang, C.L.: Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences. Sage Publications (1995)

Quality-driven Design and Management of Service-oriented Software Systems*†

Tan Phan

Faculty of ICT, Swinburne University of Technology, 3122 Hawthorn, Australia
tphan@ict.swin.edu.au

Supervised by Jun Han, Jean-Guy Schneider (Swinburne University of Technology) and Steven Versteeg (CA Labs Melbourne)

Abstract. Aligning SOA service and system properties with original business requirements during service design and operation is a major challenge that current research has not addressed in full. In this PhD work, we introduce the HOPE (High-level Objective-based Policy for Enterprises) framework that supports in a systematic manner the specification of quality-oriented policies at the business level and their refinement into policies at the system/service level. Our work is also aimed at defining an effective mechanism for business-oriented runtime monitoring of system operations and service interactions for quality conformance. Our further objective is to define adaptation mechanisms to overcome non-compliance. Our focus is on the security domain. Central to our approach is a service registry which acts as a facility for the management of policy lifecycle, to maintain the association of high-level business policies, quality objectives, and system level policies.

1 Background and Motivation

Business rules and regulations from regulatory standards such as SOX 404 [1] control the operation of many business processes and thus constrain the development and usage of IT systems that support those processes, which include Web Service(WS)-based SOA systems. While many business rules and regulations must be translated into functional requirements for such software systems, others can be translated into quality requirements, such as those concerning security, availability and manageability. These requirements can be formulated as high-level quality objectives, e.g., *"Customer data must be kept confidential"* and realized using various means of IT management and governance.

To ensure that WS-based SOA systems are interoperable and dependable, various industry standards have been proposed to support the specification and management of quality aspects of WS such as those about security, reliable messaging, and transactions. In general, these standards are about system-level

* This work is supported by the Australian Research Council and CA Labs.
† The author would like to thank Ingo Muller for his valuable comments for this paper

mechanisms used to achieve some non-functional qualities. Example mechanisms in security are role-based access control and message encryption and signing. The WS Policy framework (WS-Policy) [2] is a standard that supports the specification of quality properties for Web Services and service systems. Such standards, however, focus on describing low-level technical details for governing service interactions, rather than on specifying high-level, business-oriented requirements.

One of the issues that needs to be addressed is how to *align the high-level, business-oriented quality objectives with the system-level realization mechanisms* offered by WS standards such as WS-Policy. Currently, the high-level quality objectives are often identified by practitioners such as business analysts or IT compliance officers who often do not have an in-depth understanding of all the system-level realization mechanisms for such quality objectives. It is the system developers who are responsible for realising them. This realisation process is rather ad hoc and it is thus difficult to ensure that a system fully possesses all the required properties. As such, a contribution of great value would be a systematic process and related techniques that can *derive* the system-level realization from the business-level requirements and can verify that the realization actually fulfils the requirements.

Once we have come up with a set of design-time system-level realization mechanisms for the quality objectives, a step further is to address the issue of how to guarantee that the realization mechanisms are actually fulfilled when the system is in operation. Being able to select the appropriate quality aspects to monitor and being able to map the monitored events to the original requirements would increase the chance of identifying and resolving non-conformance. An adaptation mechanism which can analyse the quality non-conformance and derive a set of changes to be performed is needed to ensure that the requirements are always respected.

Related Work. While SOA governance is a very active research field, many of the issues identified above have not been addressed in full. Firstly, even though there exist a number of policy frameworks and languages with associated refinement and management techniques such as Ponder [3] and KAoS [4], none of them are readily applicable for specifying SOA. In particular, most of the existing frameworks fall short in enabling the specification of business requirements and the refinement of them into system-level policies. More details can be found in our review paper in [5].

There have been a number of attempts to apply model-driven architecture (MDA) techniques for the modelling and translation of SOA qualities into system-level realization mechanisms such as [6] or [7]. In such work, quality properties of services and applications are modelled in platform-independent manners which are then transformed into platform-dependent codes and configurations for middle-wares to realize these qualities. However the entities being modelled are technical entities, representing technical concepts like *filter*, *connector*, *services*, and *proxies*, not business-oriented entities. This not only limits the participation of business analysts and IT compliance officers in the modelling process but also makes it hard to align the models with the original business requirements.

It is also seen that even though there are different approaches for WS management, the support for business-oriented management is not adequate. Current research work in the field of runtime service management focuses more on the system-level management. Various techniques have been proposed for the specification of service properties and the monitoring and management of them such

as in WSLA [8], SLANG [9], and WSMN [10]. However, little has been done to allow for the management of services from a business perspective even though the importance of business-oriented service management has been acknowledged such as in [11]. The main focus of work in this area has been on specifying and enforcing service SLA. We are unaware of any work that attempts to incorporate other business aspects such as compliance to standards, rules and regulations.

Research objectives. The main objectives and primary contributions of this thesis are as follow 1) We define a general framework for the specification of high-level quality requirements for systems and present a mechanism to refine them to system-level realization mechanisms 2) We provide an approach for policy-based service registration and discovery with algorithms and techniques for detecting non-compliance of services to organizations' quality requirements and for verifying service-client quality requirement compatibility 3) We aim to provide a novel monitoring mechanism that can map service runtime interactions back to the original business requirements in an intuitive manner. For this objective, we consider the use of techniques such as Finite State Automata for modelling the requirements and Bayesian network for failure analysis 4) We aim to define techniques and algorithms for generating actions that can be performed to guarantee compliance to quality requirements. We consider applying techniques in the field of Autonomic computing for this. For all of these, our focus is on the alignment of business requirements and SOA systems and we use security as the example domain for our approach.

2 The Approach and the HOPE Framework

The approach. We propose the High-level Objective-based Policy for Enterprises (HOPE) framework which is aimed at addressing the above research problems via policy-based design and management. As presented in Figure 1, with HOPE, we specify quality requirements (which are driven by original business `rules and regulations`) in the form of business-level `policies` which will then be formulated as `quality objectives` applicable on business entities. The `objectives` are then refined into *system-level* `Web Services policy` for Web Services-based applications. Such policies are used to regulate runtime service interactions and provide information for adaptation in case of policy non-compliance.

Central to the approach is a `service registry` which acts as a facility for the management of policy lifecycle including the *modelling, analysis and design*, *creation*, *usage*, *update*, *removal* of policies; and maintain the association of high-level business `policies`, `quality objectives`, and `system-level policies`. The *registry* provides a point of reference for various design time and runtime management operators such as *WS-Monitor and WS-Enforcer* to retrieve policy-related information and store the relevant data that they collect.

Our approach employs policy-based management which has the advantages of being able to dynamically update the behaviour of a managed system according to the changing context requirements without having to modify the implementation of the managed system. Also, the declarative specification of rules and regulations in the form of policy statements are more concise, intuitive and simpler to verify than

procedural code. Furthermore, service registries hold service metadata and are characterized by rich metadata management and rich query capabilities. As policy is one important type of SOA metadata, service registries' capabilities can be extended for policy-based management.

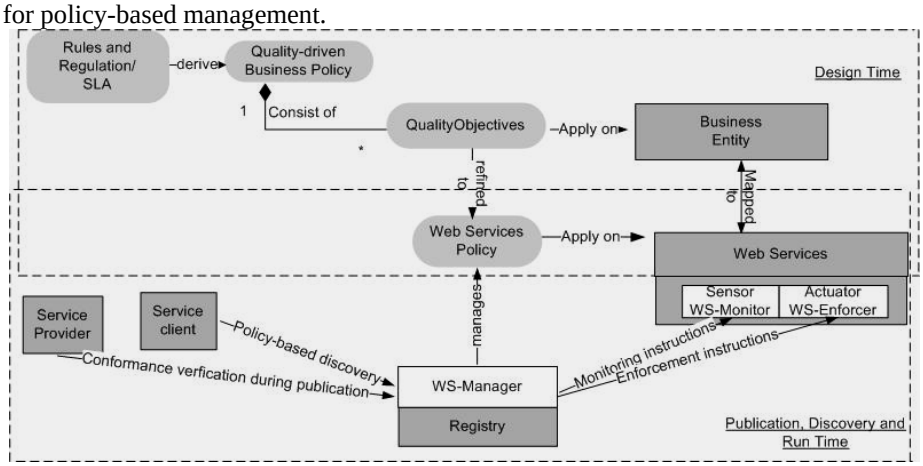

Fig. 1. Policy-based SOA quality management using a service registry

Validation of the approach. To validate the work we use a business case study together with research prototyping. We design a business case scenario and examine a database of global rules and regulations from the Unified Compliance Framework (http://www.unifiedcompliance.com/) to identify applicable business requirements. We then, from such rules and regulations, using our approach to derive business policies and use such policies to validate our refinement, monitoring and adaptation techniques. A prototype for HOPE is also being built and once the tool is ready, we plan to present it to a group of developers and business analysts for validation.

Work to date

During the past 18 months, I have worked on addressing a number of research issues, aligning with research objectives (1) and (2) presented above. Details are as follows.

Quality-oriented business policy specification and refinement. We investigated the research issue of quality-driven business policy specification and refinement of SOA Systems in [12]. In this work, we proposed a framework (Figure 2) that supports the specification of business level quality-oriented policies and their refinement into policies at the system/service level.

As can be seen in Figure 2, in our approach, quality-oriented business requirements (`quality requirements`) are expressed as `quality objectives` applied to business entities which are modelled in `application entity model`. These `objectives` are then refined or translated into *system-level* `WS-Policy statements`. The refinement relies on an application-specific `business entity model` and application-independent `domain quality models`, for which we created the meta-models. We illustrated the approach with a *Mortgage loan approval* business case study to demonstrate the policy specification and refinement for qualities in the security domain and have implemented a proof of concept prototype.

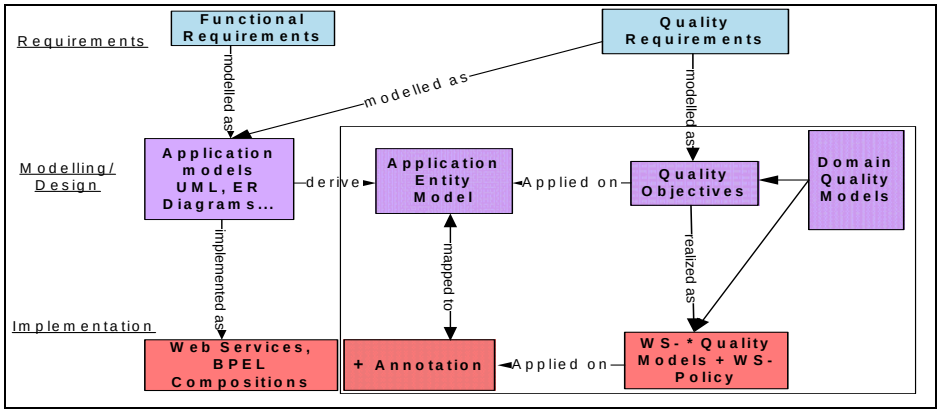

Fig. 2. A framework for quality-driven policy specification and refinement

Policy-based service registration and discovery. Another research issue that we have identified and addressed was the issue of policy-based registration and discovery [13]. In [13], we argued that (1) ensuring service qualities (specified in the form of WS polices) are consistent with organizations' regulations and (2) matching service and client policies for effective service discovery are issues yet to be addressed. We thus presented a new approach (Figure 3) that allows for the automatic verification and matching of policies, using a `service registry.` The registry serves as a policy storage and management facility, a policy checkpoint during *service publication*, and as a policy matchmaker during *service discovery*. We extended WS-Policy with a `policy conformance` algorithm for policy verification at *service publication time* and used `WS-Policy Intersection` for policy matching at *service discovery time*. We have developed a policy information model and the policy processing capabilities for the registry. A prototype has also been implemented.

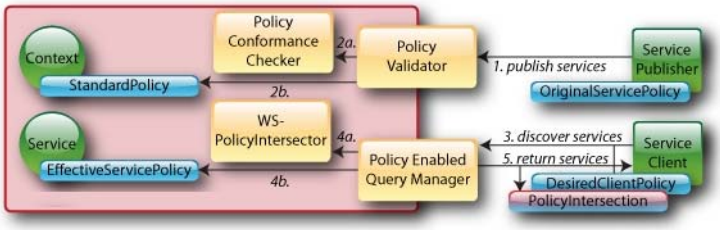

Fig. 3. A registry-centric model for policy-based service registration and discovery

4 Conclusion

We have presented in this paper a discussion about the limitations of current approaches in business-oriented design and management of qualities for SOA systems and outlined a framework for addressing this issue. Our approach is aimed at aligning business-oriented rules and requirements with system-level management via a mechanism that allows for the specification of quality-oriented business rules and

regulations and the refinement of them into system-level quality. We also provide a general mechanism that utilizes a service registry for quality-based service registration and discovery. Our future work is on monitoring techniques that can relate monitored service interactions to the original high-level business requirements and a business-oriented adaptation mechanism for overcoming non-compliance.

Reference

1. Sarbanes, P.: Sarbanes-Oxley Act of 2002. The Public Company Accounting Reform and Investor Protection Act. Washington DC: US Congress (2002)
2. Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Langworthy, D., Malhotra, A.: Web Services Policy Framework (WS-Policy). Version **1** (2006) 2006-2003
3. Nicodemos, D., Naranker, D., Emil, L., Morris, S.: The Ponder Policy Specification Language. Proceedings of the Int'l Workshop on Policies for Distributed Systems and Networks. Springer-Verlag (2001)
4. Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S., Lott, J.: KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement. IEEE 4th Int'l Workshop on Policies for Distributed Systems and Networks, 2003.POLICY 2003. (2003) 93-96
5. Phan, T., Han, J., Schneider, J.G., Ebringer, T., Rogers, T.: A Survey of Policy-Based Management Approaches for Service Oriented Systems. 19th Australian Conf on Software Engineering, 2008. ASWEC 2008 (2008) 392-401
6. Wada, H., Suzuki, J., Oba, K.: A Model-Driven Development Framework for Non-Functional Aspects in Service Oriented Architecture. 2006 Int'l Conf on Autonomic and Autonomous Systems (ICAS 2006). IEEE Computer Society, Silicon Valley, California, USA (2007)
7. Comerio, M., Paoli, F.D., Grega, S., Maurino, A., Batini, C.: WSMoD: a Methodology for Qos_based Web Services Design. Int'l Journal of Web Services Research **4** (2007) 28
8. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management **11** (2003) 57-81
9. Lamanna, D.D., Skene, J., Emmerich, W.: SLAng: A Language for Defining Service Level Agreements. Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems-FTDCS (2003) 100-106
10. Sahai, A., Machiraju, V., Sayal, M., van Moorsel, A., Casati, F.: Automated SLA Monitoring for Web Services. IEEE/IFIP DSOM **2002** (2002)
11. Casati, F., Shan, E., Dayal, U., Shan, M.-C.: Business-oriented Management of Web Services. Communication of the ACM **46** (2003) 55-60
12. Phan, T., Han, J., Schneider, J.-G., Wilson, K.: Quality-Driven Business Policy Specification and Refinement for Service−Oriented Systems. Int'l Conf on Service Oriented Computing (ICSOC 2008). Springer, Sydney, Australia (December 2008) 17 pages, to appears
13. Phan, T., Han, J., Schneider, J.-G., Ebringer, T., Rogers, T.: Policy-based service registration and discovery. In: Meersman, R., Tari, Z. (eds.): Int'l Conf on Cooperative Information Systems 2007 CoopIS 2007, in On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. Springer, Villamoura, Algarve, Portugal (2007) 417-427

Supporting Documentation and Evolution of Crosscutting Concerns in Business Processes

Chiara Di Francescomarino
supervised by Paolo Tonella

dfmchiara@fbk.eu - Fondazione Bruno Kessler, Trento, Italy

Abstract. Business processes can be very large and can contain many different concerns, scattered across the process and tangled with other concerns. Crosscutting concerns are difficult to find, locate and modify in a consistent way, thus making process maintenance and reuse hard, even for business experts.
In this work, we address the problem of supporting business designers when they need to document existing crosscutting concerns and when they work on their evolution. We propose to add semantic annotations from a domain ontology to BPMN process elements in order to be able, on their basis, to manually or semi-automatically mine concerns crosscutting the process. Once modularized separately from the principal process flow, we aim at supporting the consistent evolution of all the process parts they crosscut, by means of a BPMN-based rule language.

1 State of the Art

Business processes can be very large and involve many different perspectives. Business process (BP) description and modelling languages provide means for describing the main views which define the principal decomposition of the business process, such as the control flow, data flow and performers. Though allowing to represent these perspectives, the available languages are not able to capture all different types of relevant features in the process, above all if scattered across the process and tangled with one another. This lacking capacity in identification and explicit representation of crosscutting concerns (CCs) results not only in difficulties during process comprehension, but also in a challenging maintenance and evolution phase.

In order to improve process flexibility, to make maintenance and evolution easier and to improve reuse, the Business Process Community made several attempts, such as flexible and dynamic processes and generic processes [1, 2].

An aspect is a module composed by *pointcut designators* and *advices*. A pointcut provides a condition to specify a set of so called *join points*, i.e., precise points in the execution flow that the aspect intercepts, by means of some quantification mechanism. An advice, instead, specifies how to realize the concern and when in the join points of interest the concern execution has to be activated, thus realizing the AOP (Aspect Oriented Programming) *obliviousness* (the primary program ignores advices) [3].

Recently, AOP has investigated the migration of existing systems to aspects. Such transformation process involves at least two steps: (1) identification of scattered and tangled concerns (i.e. aspect mining); (2) migration of crosscutting concerns into actual aspects (i.e. aspect refactoring). A number of different techniques have been developed to (partially) automate aspect mining. Some of them rely on Formal Concept Analysis (FCA).

The AOP approach has already been applied to a specific process executable description language, BPEL (Business Process Execution Language). AO4BPEL [4] is a dynamic aspect oriented extension of BPEL, enriched with concepts for describing pointcut designators (XPath expressions to locate the places where the aspect is applied) and advices (fragments of BPEL code). A similar approach, mainly differing from AO4BPEL in the advice language has been proposed by Verheecke et al. [5]. Padus [6] is yet another aspect-oriented BPEL extension, but without dynamic weaving and improving portability.

All the cited works mainly focus on the developers' perspective, without considering that, in practice, business designers prefer a higher level modelling notation, such as BPMN (Business Process Modelling Notation). Moreover, explicit decomposition of the process into totally separate modules for the aspects, may hinder, instead of simplifying, process design and comprehension.

In order to address the SoC (Separation of Concerns) problem in general purpose code, several tools (e.g. JQuery [7]) support programmers in source code navigation and concern localization. We propose a similar approach for BPs. In order to help business experts to understand and maintain BPs, we introduce a visual language for querying BP models. However, since the raw syntactical information available in BPMN diagrams is not enough for characterizing the concerns being mined, we add semantic expressiveness to models by means of semantic annotations.

BP semantic annotation techniques have already been proposed by several authors (e.g. [8]), for different process description languages and addressing different issues. The main purpose of semantic annotations in this work is another one. By allowing to manually (via the query language) or semi-automatically (via FCA) mine for business domain CCs and to reason about domain ontology concepts, they aim to support concern documentation and domain constraint compliance.

This work supports business experts in BP crosscutting concern documentation and evolution, thus contributing to process understandability, maintainability and evolvability. In order to achieve this goal, we devise a four-fold contribution to the state of the art: (1) we propose to enrich BP models with semantic annotations and suggest to support the business designer both in the annotation phase, by suggesting correct annotations or verifying user-defined ones, and in the related domain ontology building and/or enrichment activity; (2) we define an easy-to-use, visual query language for querying BPs and we support the business user in formulating queries and visualizing the results; (3) we investigate a technique for the automatic mining of candidate CCs in BPs; (4) we propose a

method to document CC evolution and, whenever any CC changes, to update all modules it crosscuts, in an automatic or semi-automatic way.

The paper is organized as follows: in Section 2 we give an overview of our research directions; in Section 3 we detail one of such directions (the visual query language for exploration and documentation of CCs in BPs); finally, conclusions and future works are presented in Section 4.

2 Crosscutting Concern Documentation and Evolution

Although BPMN allows the explicit representation of the main views of a process, it often leaves implicit the crosscutting features. Knowledge about the crosscutting functionalities pertains to the BP semantic (vs. syntactic) description, but "BPM doesn't provide a uniform representation of an organization's process space as a whole on a semantic level which would be accessible to intelligent queries and inference" [8].

Business Process Semantic Annotation. In order to supply BPs with semantic information from the business domain, we propose to semantically annotate them with concepts taken from a business domain ontology. For the semantic annotation of BPMN elements, we use a standard BPMN textual annotation, augmented with an "@" before the semantic concept. Such annotations open to the possibility of reasoning about semantic properties, thus facilitating the compliance with possible predefined constraints, the process of manual or semi-automatic crosscutting concern mining and the consistent update of all the modules crosscut by a modified documented feature.

An example of a semantically annotated BPMN diagram is shown in Figure 2. It represents a process for realizing an assembled product starting from three raw products. All the flow objects in the process are annotated with concepts taken from a "buying and selling" domain ontology.

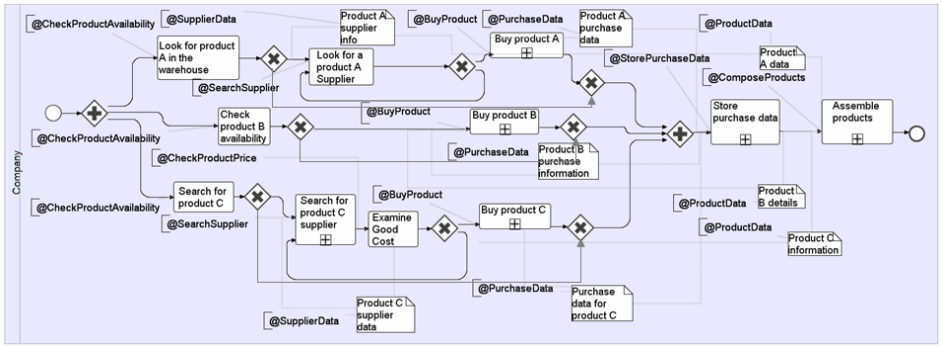

Fig. 1. An example of a semantically annotated BPMN process.

Clearly business designers need to be supported in the semantic annotation phase. On one hand they can be supplied with annotation suggestions compliant with possible predefined constraints and, on the other, mechanisms for the automatic compliance verification of user defined annotations with constraints can be provided. Moreover, they will also be supported in ontology building or enrichment by means of dedicated tools and techniques. We plan to use an ad-hoc linguistic analysis for ontology extraction from process entity labels, since available ontology learning techniques are not applicable to small textual corpora such as those represented by process labels.

BPMN Visual Query Language. In order to support crosscutting concern documentation, understanding and evolution, we propose the use of a query language able to quantify on the BP elements, thus localizing the concerns of interest. By carefully considering usability of the query language (it is supposed to be used by business designers), we chose a language as close as possible to what business experts already know: BPMN itself. We slightly extended BPMN to BPMN VQL (Visual Query Language), so as to make it usable for query purposes as well. BPMN VQL is described in more detail in Section 3.

Concern Mining. The support provided by query languages and concern browsers is of fundamental importance, but it still involves substantial effort from the users, thus breaking the ease-of-use requirement of business designers. It can be complemented by an automated concern mining tool, which automatically suggests candidate CCs to designers, both structural (e.g. workflow patterns) and business domain oriented (e.g. purchasing activities). In order to achieve this goal, we investigated novel aspect mining techniques for BPs, based on FCA (more details in [9]).

Crosscutting Concern Aspectization. Once CCs are known and their existence is documented (through a collection of queries), their evolution too has to be supported and documented. Changing a crosscutting concern, due to its scattered and tangled nature, is not a trivial job. The update has to be applied in a consistent way to all the concern occurrences and the tangled subparts. Manually executing the change is a tedious and error-prone task. In order to support business designers, we propose to use a BPMN-based rule-language, that extends BPMN VQL with an "update" part in order to support BP manipulation at selected points. The rule could be either weaved and stored, in order to only document the concern changes (and potentially re-weaved whenever the designer wants) or it can be kept alive during workflow editing, in order to detect any possible violation of the rule, giving rise to a warning and a (semi-)automated fix (more details in [9]).

3 BPMN Visual Query Language

The exploration of large BPs and the retrieval of interesting CCs can be a difficult and time consuming task for business experts. In order to support them, we propose a query language able to quantify over the BP elements, localize interesting concerns and, once identified, present them to users by visually high-

lighting their occurrences in the process diagram. Since a critical issue of the query language is usability, the proposed language is a *visual* language based on BPMN, the standard language business designers use for representing BPs.

The BPMN Visual Query Language (BPMN VQL) allows to formulate queries by using standard BPMN graphical notation and new specific operators. Moreover, the language provides a different notation to distinguish between the "matching" part of the query, that determines the criterion to match, and the "selection" part (characterized by darker background, thicker lines and bold font style), that allows to visualize only the selected subpart of the matching result.

In the following subsection we present some examples of uses of the BPMN VQL referring to the product assembly process shown in Figure 2. In order to formalize the queries and give them a precise semantics, we provide their translation into SPARQL, an RDF-based query language More details about the language can be found in our Technical Report ([9]).

3.1 BPMN VQL Use Cases.

The BPMN VQL can be used to query for any concern of interest that is part of the business process design. A simple example is provided in Figure 2.

For particularly relevant concerns, it makes sense to store the query as part of the design artefacts and to keep it as a precise documentation of the design choices related to the specific concern. For example, the business designer, in order to be compliant with specific privacy laws or business strategies, could be interested in storing the concern related to the product supplier search management as a relevant design feature. The query is shown in Figure 3.

Moreover, availability of design artefacts documenting the relevant concerns of a BP is particularly useful when the process design evolves, due to new customer requests or to environmental changes.

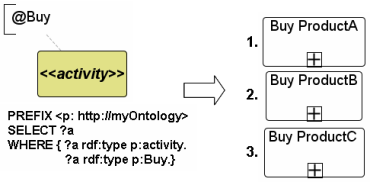

Fig. 2. Query asking for all the activities (both tasks and subprocesses) that buy something (left) and query result (right).

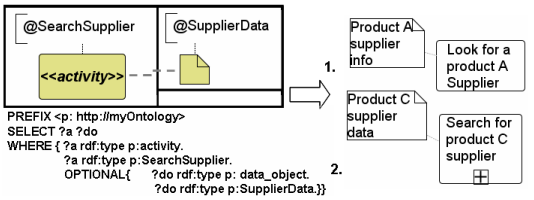

Fig. 3. Query asking for all the activities that search suppliers and, if any, the "SupplierData" data objects they produce (left). On the right the query results.

4 Conclusion and future works

We aim at realizing a framework able to support business users in BP CC documentation and evolution. Up to now we have: (1) studied and partially imple-

mented techniques both for semi-automatic semantic annotation of BPs and for domain ontology building and enrichment; (2) defined a BPMN VQL for querying BPs; (3) investigated, adapted and developed approaches for semi-automatic mining of business domain CCs; (4) studied rule languages for CC evolution.

In the future we will complete the implementation of tools for the user support in BP semantic annotation and domain ontology building; in the visual querying of business processes; in the semi-automatic mining of CCs; and in their evolution. Moreover, we will continue our research in the following directions: (1) linguistic analysis and normalization techniques, aimed at ontology construction; (2) ontology reasoning and constraint definition and compliance; (3) FCA analysis criteria, to provide more automation during the identification of concern candidates; (4) new mining techniques, based on structural properties of the workflow; (5) weaving and re-weaving of aspectized concerns; (6) validation of the benefits gained by applying the suggested methodologies to case studies involving business users; (7) extension to the process execution phase.

References

1. van der Aalst, W.M.P.: Generic workflow models: How to handle dynamic change and capture management information? In: Conference on Cooperative Information Systems. (1999) 115–126
2. A. Schnieders, F.P.: Variability mechanisms in e-business process families. In: 9th International Conference on Business Information Systems (BIS 2006), W. Abramowicz, H. Mayr (2006) 583–601
3. Filman, R.E., Elrad, T., Clarke, S., Aksit, M.: Aspect-Oriented Software Development. Addison-Wesley (October 6, 2004)
4. Charfi, A., Mezini, M.: Aspect-oriented web service composition with AO4BPEL. In: Proceedings of the 2nd European Conf. on Web Services (ECOWS). Volume 3250 of LNCS., Springer (2004) 168–182
5. Verheecke, B., Cibràn, M.A., Jonckers, V.: Aspect-oriented programming for dynamic web service monitoring and selection. In Zhang, L.J., ed.: ECOWS. Volume 3250 of LNCS., Springer (2004) 15–29
6. Braem, M., Verlaenen, K., Joncheere, N., Vanderperren, W., Straeten, R.V.D., Truyen, E., Joosen, W., Jonckers, V.: Isolating process-level concerns using padus. In Dustdar, S., Fiadeiro, J.L., Sheth, A.P., eds.: Business Process Management. Volume 4102 of LNCS., Springer (2006) 113–128
7. Janzen, D., De Volder, K.: Navigating and querying code without getting lost. In: AOSD '03: Proceedings of the 2nd int conf on Aspect-oriented sw development, New York, ACM Press (2003) 178–187
8. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: a vision towards using semantic web services for business process management. (2005) 535–540
9. Tonella, P., Di Francescomarino, C.: Business process concern documentation and evolution. Technical report, Fondazione Bruno Kessler (FBK-IRST), Trento (2008) http://se.fbk.eu.

Pervasive Services Engineering for SOAs

Dhaminda Abeywickrama
(supervised by Sita Ramakrishnan)

Clayton School of Information Technology, Monash University, Australia
dhaminda.abeywickrama@infotech.monash.edu.au

Abstract. With the proliferation of ubiquitous computing devices and mobile internet, it is envisaged that future pervasive services will be increasingly large-scale and operate at an inter-organizational level. Designing and implementing pervasive services will therefore become a more complex and challenging task. Significant interest exists within the pervasive computing community for representing pervasive services at different stages of the software life cycle. While most of these efforts have focused on the detailed design or implementation of pervasive services little work has been done at the software architectural level. In this research, we propose a novel approach based on behavioral modeling and analysis techniques for representing pervasive software services and their compositions and verifying the process behavior of these models against specified system properties. This systematic, architecture-centric approach combines the benefits of principles such as UML, model transformation of Model Driven Architecture, and formal behavioral modeling and analysis techniques through model-checking, for engineering pervasive software services. In order to illustrate the validity and practical feasibility of the proposed approach we use an existing case study in transport and logistics. The approach will be evaluated to demonstrate the effectiveness of model-checking as a technique for verifying pervasive software services.

1 Introduction: Motivation and Research Objectives

Mark Weiser's vision of ubiquitous computing has been the impetus behind the introduction of a new service-oriented computing paradigm known as pervasive services or context-aware services. Ubiquitous environments facilitate the collection of information from various data sources in order to aggregate the context of entities, such as users, places or objects. The context obtained from these sources can be used to automatically adapt a service's behavior or the content it processes to the context of one or several parameters of a target entity in a transparent way, resulting in pervasive services [1]. With the advancement of ubiquitous computing devices and mobile internet, it is envisaged that future pervasive services will be large-scale and operate at an inter-organizational level, with an increasing number of actors and constraints involved [2]. The real-time requirements, highly dynamic nature, quality of context information and automation further contribute to making pervasive services complex and challenging compared to conventional services. Thus, the design and implementation of

pervasive services will be a more complex task. Significant interest exists within the pervasive computing community for representing pervasive services at different stages of the software life cycle. However, so far most of the research work has been focused on the detailed design or implementation stages [3, 4] of the software life cycle such as Web services while little attention has been given to the initial phase of design such as architecture design, providing the motivation for this research.

The use of models has been a popular approach for engineers when constructing complex systems. Behavior modeling and analysis have been successfully used by software engineers to uncover errors of concurrent and distributed systems at design time. In this research, we propose a novel approach based on behavioral modeling and analysis techniques for modeling pervasive software services and their compositions and verifying the process behavior of these models against specified system properties. This systematic, architecture-centric approach combines the benefits of principles such as UML, model transformation techniques of Model Driven Architecture (MDA), and formal behavioral modeling and analysis techniques using model-checking, for engineering pervasive services (research objectives: Fig. 1a). Context-handling information is considered to be tightly coupling or crosscutting the core functionality of a service at service interface level [5]. In this research, the crosscutting context-dependent functionality of the interacting pervasive services is modeled as aspect-oriented models in UML. In order to facilitate formal behavioral analysis, these UML models are automatically translated to state machine based behavioral representations using transformation authoring of MDA. The behavioral modeling and analysis approach used in this research is particularly based on formal verification, validation and simulation techniques provided by the model-checker, the Labeled Transition System Analyzer (LTSA) [6] and its process calculus Finite State Processes (FSP). Model-checking is a formal verification approach for verifying finite state concurrent systems. We use an existing case study in transport and logistics to explore the approach and to illustrate its practical feasibility. The approach will be evaluated to demonstrate the effectiveness of model-checking as a technique for detecting design defects in pervasive service specifications.

Preliminary results of this research have been published in [7, 8]. While this paper discusses the overall research in general it particularly reports on the model transformation tool created using IBM Rational Software Architect 7.0 [9], and further research directions established on formal verification and validation of the research. Section 2 provides a brief analysis of the related work. In Sect. 3, the overall research methodology of the study is discussed, and Sect. 3.1 and 3.2 present the model transformation tool created, and model verification steps proposed as future work, respectively. Finally, Sect. 4 concludes the paper.

2 Related Work

Previous approaches to the development of pervasive services have largely been at the detailed design or implementation stages [3, 4] of the software life cycle.

Luo et al. [3] establish a framework that enables context-aware composition of Web services taking into account both the user's and the service's context when composing services. In [4], the authors propose an approach to include context in the composition of Web services. However, the approach taken in this research is clearly distinctive from the above approaches as it is based at the software architectural level which is higher level and abstract design. A similar approach to this, where concurrent behavior between base programs and aspects has been modeled using FSP semantics is provided in [10]. However, our work differs significantly as it is based on pervasive services. In [5], the authors present an approach on model driven development of pervasive services using UML and models crosscutting pervasive concerns as aspects. However, they have taken no account of concurrency and distributed notions in their design or any formal verification aspects through techniques such as model-checking as proposed in this research.

3 A Methodology for Pervasive Services Engineering

In this section, we provide an overview of the research methodology applied for engineering pervasive software services. Next we present the model transformation tool implemented using IBM Rational Software Architect, and further research directions established on model verification and validation, as part of the research methodology. The research approach is explored using a real-world case study in intelligent tagging for transport and logistics called the ParcelCall project [11], which is a European Union project within the Information Society Technologies program. The case study describes a scalable, real-time, intelligent, end-to-end tracking and tracing system using radio frequency identification, sensor networks, and services for transport and logistics. A significant subset of the ParcelCall case study is exception handling that needs to be enforced when a transport item's context information violates acceptable threshold values. This research is focussed on this subset.

The overall pervasive service-oriented development process is divided into three main stages (Fig. 1b). First, using the case study subset, a service specification for the system under consideration is defined. For this purpose, the relevant use cases for the case study subset are determined and the services that realize the identified use cases are specified. The identified use cases and their relationships are expressed using an UML use case diagram. As defined by Kruger et al. [12], we identify a software service as the interaction pattern or the interplay of several components collaborating to complete a desired task. Furthermore, we identify services as first-class modeling elements as opposed to first-class implementation elements, such as Web services. Message sequence charts (MSCs) provided by the LTSA-MSC tool, which is an extension to the LTSA tool, are used to describe the interaction patterns defining the services for the case study subset.

Second, the architecture for the system under consideration is defined. To this end, first a component configuration that implements the extracted services

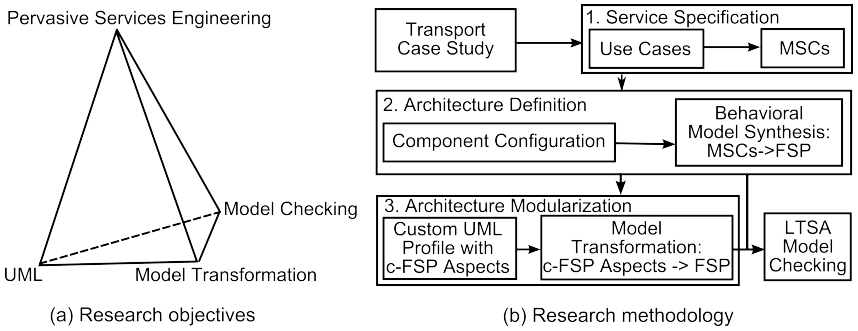

Fig. 1. Research objectives and methodology.

is defined using an UML deployment diagram. The component configuration of the approach is based on a distributed version of the Observer pattern called the Event-Control-Action architecture pattern [13]. Second, a behavioral representation of the service specification in the form of FSPs (architecture model) is generated automatically using the LTSA-MSC tool's FSP synthesis feature. The architecture model provides the basis for modeling and reasoning about the system design where the components are modeled as labeled transition systems.

Third, the architecture model synthesized in the previous step is modularized by applying separation of concerns. Context-handling information is considered to be crosscutting the core functionality of a service at service interface level, which results in a complex design that is hard to implement and maintain. Therefore, a custom UML profile is proposed from which a UML model template and aspect-oriented models in UML (contextual-FSP or c-FSP aspects) [8] are derived for the case study subset. A custom prototype tool has been implemented to automate the translation of the c-FSP aspects to formal FSP to facilitate rigorous verification by the LTSA. Details of this tool implementation and the proposed model verification process using the LTSA are discussed next.

3.1 Model Transformation

The custom prototype tool has been built using the Java Emitter Templates (JET) transformation authoring feature of IBM Rational Software Architect (7.0) [9]. JET is an open source technology developed by IBM and is part of several IBM Rational modeling platform 7.0 products. The transformation is used to automate the application of design patterns and generate infrastructure code for the c-FSP aspects using FSP semantics (aspectual FSP generation). An added benefit of applying JET transformation is that it allows the end-user modification of the code generator using templates. The application of model driven development in pervasive services engineering at state machine level is novel. The main benefits of this approach are improving the quality and productivity of service development, easing system maintenance and evolution, and increasing the portability of the service design. Two variations of the model transformation tool

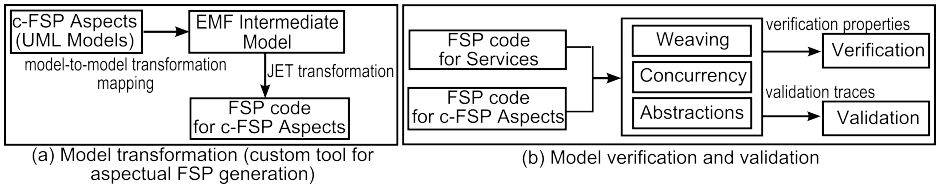

Fig. 2. Model transformation, model verification and validation.

have been built. Initially, a model-to-text transformation was implemented with XPath expressions to navigate the UML models for c-FSP aspects and extract model information dynamically to the transformation. However, JET's support for UML models has two main limitations. First, using JET it is not possible to access stereotype information of the UML models although custom XPath functions can be written to perform this, and second JET authoring finds the complexity of the UML meta-model hard to manage. Therefore, a more effective solution was implemented by first creating a model-to-model mapping transformation which extracts relevant stereotype information from the UML models, and builds a code-generator specific intermediate Eclipse Modeling Framework model which can then be consumed by the JET transformation. Thus, a multi-stage transformation (Fig. 2a) has been used to transform the UML models to formal FSP semantics.

3.2 Model Verification and Validation

This section presents a discussion on the proposed future work of this research. For model verification (Fig. 2b), first the generated FSP for the c-FSP aspects need to be woven with their base state machines using an explicit weaving mechanism. This can be modeled as the parallel composition of the aspectual and base state machines in FSP. Synchronization events can be introduced to control the coordination and weaving order between the base program and aspects. Concurrency and distributed notions will be added to the interacting pervasive software services and their compositions, such as shared objects, message passing and concurrent architectures. In addition, proper abstraction mechanisms need to be enforced on the models as part of preparation for verification. Formal model-checking techniques provided by the LTSA, such as safety, progress and absence of deadlocks, will be used to verify the process behavior of the services against system properties specified to extensively cover the system requirements. The LTSA checks for property violations and if any violations are found it produces a trace to the violation known as a counterexample, which can be used to improve the state models or the system properties for the pervasive services. Validation of the models will be performed using the simulation and animation features of the LTSA. The research approach will be evaluated to demonstrate the effectiveness of model-checking as a technique for verifying pervasive software service specifications. Potential defects can be injected to the service specification and results can be compared with the defect free service specification.

4 Conclusion and Future Work

To summarize, the primary objective of this research is a systematic, architecture-centric approach for engineering pervasive services based on the principles of UML, model transformation techniques and formal model-checking. Building software architectural models of pervasive services provide engineers a better understanding of how these complex services inter-operate. Also, it helps to uncover any errors during the early stages of the software life cycle before the services are actually implemented, thus contributing to the Web services community. Future work involves model verification and validation as discussed previously.

References

1. Hegering, H. G., Kupper, A., Linnhoff-Popien, C., Reiser, H.: Management Challenges of Context-Aware Services in Ubiquitous Environments. In: Goos, G., Hartmanis, J., van Leeuwen, J. (eds.) DSOM 2003. LNCS, vol. 2867, pp. 246–259. Springer, Heidelberg (2003)
2. Buchholz, T., Kupper, A., Schiffers, M.: Quality of Context: What It Is And Why We Need It. In: 10th Workshop of the HP OpenView University Association (HPOVUA'03). (2003)
3. Luo, N., Yan, J., Liu, M., Yang, S.: Towards Context-Aware Composition of Web Services. In: Fifth International Conference on Grid and Cooperative Computing. pp. 494–499. (2006)
4. Mostefaoui, S. K., Hirsbrunner, B.: Context-Aware Service Provisioning. In: IEEE/ACS International Conference on Pervasive Services. pp. 71–80. (2004)
5. Sheng, Q. Z., Benatallah, B.: ContextUML: a UML-based Modeling Language for Model-Driven Development of Context-Aware Web Services. In: International Conference on Mobile Business (ICMB'05). pp. 206–212. (2005)
6. Magee, J., Kramer, J.: Concurrency: State Models and Java Programs, 2nd Edition. John Wiley and Sons, Worldwide Series in Computer Science (2006)
7. Abeywickrama, D., Ramakrishnan, S.: A Model-Based Approach for Engineering Pervasive Services in SOAs. In: 5th ACM International Conference on Pervasive Services (ICPS'08). pp. 57–60. (2008)
8. Abeywickrama, D., Ramakrishnan, S.: Towards Engineering Models of Aspectual Pervasive Software Services. In: 3rd ACM Workshop on Software Engineering for Pervasive Services (SEPS'08). pp. 3–8. (2008)
9. IBM Rational Software Architect Transformation Authoring, `http://publib.boulder.ibm.com/infocenter/rsdhelp/v7r0m0`
10. Douence, R., Le Botlan, D., Noye, J., Sudholt, M.: Concurrent Aspects. In: 5th International Conference on Generative Programming and Component Engineering (GPCE'06). pp. 79–88. (2006)
11. Davie, A.: Intelligent Tagging for Transport and Logistics: The ParcelCall Approach. Electronics and Communication Engineering Journal 14(3), 122–128 (2002)
12. Kruger, I. H., Mathew, R., Meisinger, M.: Efficient Exploration of Service-Oriented Architectures using Aspects. In: 28th International Conference on Software Engineering. pp. 62–71. (2006)
13. Costa, P. D., Pires, L. F., van Sinderen, M.: Architectural Patterns for Context-Aware Services Platforms. In: 2nd International Workshop on Ubiquitous Computing. pp. 3–19. (2005)

Towards Adaptive Service Development

Aries Tao Tao
supervised by Jian Yang
{tao, jian}@ics.mq.edi.au

Computing Department, Macquarie University, Australia
http://www.mq.edu.au

Abstract. In the dynamic e-Business environment, it is desirable for a service to meet the requirements of different users. The current available technologies rather supports a service with a single fixed business process without considering user needs. Such design makes it difficult for user to integrate the provided service, hence obstruct the service provider to expand the business. In this paper we proposed an alternative service design method - Adaptive Service Design. Inspired by the concept of *Abstraction* and *Polymorphism* in Object Oriented Computing, service adaptation allows an Abstract Business Process class to be configured by Policy and user required interface, hence dynamically generate multiple business processes to meet different user interaction requirements.

Key words: service differentiation, service adaptation, matchmaking

1 Introduction

While improvement in Service Oriented Computing (SOC) has proved effective for integration at lower levels (e.g. wrap traditional software component up to be service), service integration through service interface is till a challenge due to the heterogeneity of service specification. Service interface is a specification for user (client program) to interact with the service. It is supported by underlying business process(es), and consisted of activities which are implemented by corresponding tasks of the business processes.

In e-business world, it is common for a set of service consumer with similar service requirements to have different interfaces. The differences could be caused either by user contexts [1] **(1) at business level** such as different physical locations, market conditions, policy regulations, competitive threats, or by service implementation **(2) at application level**s such as different message formats or sequences. All of these situations and more drive the need for a service design which enable the service provider to respond quickly in support of diverse service users' requirements.

On the other hand, ignoring the user heterogeneity, the current service is designed to be context free which supports the fixed monolithic business process and interface to all users. This inflexible design makes it difficult for service consumer to perform service integration with the provided service, hence obstruct

the service provider to expand the business. SOC research community has been aware of the issue. The solutions are proposed which mainly use process matchmaking [2] to identify the mismatch patterns, and adapter [3] to overcome the mismatch. However, as functionality of adapter is to split/combine messages or swap message sequences, such solution only resolve the certain mismatch at application level.

To overcome the problem with current service design, we propose a new design that supports policy negotiation, process modification, and interface alternation according to use needs, hence cater service to quickly adapt to diver service consumer requirements. We refer this design as Adaptive Service Design (ASD). An *User Adaptation* mechanism is developed based on ASD which enables the service provider to:

- identify user functionality and interaction requirements from the user's interface.
- dynamically generate the business process and service interface to meet the specific requirements.

The basic design philosophy of ours, and one that distinguishes us from others, is that the ASD supports service provider to easily vary the policy, business process and service interface to dynamically adapt different user integration requirements in terms of functionality and interaction patterns.

This paper is organized as follows: Section 2 discusses the related work. The structure of *Adaptive Service Design (DSD)* are specified in Section 3. We finally concludes our work in section 4.

2 Related Work

The related work can be divided in following areas: (1) *Service Description* which focuses on enriching the service interface for user accessibility; (2) *Service Differentiation* which supports services to provide different functionalities to users; (3) *Service Matchmaking* which allows the mismatches between provided service and user required services to be identified; (4) *Service Adaptation* overcomes the mismatches identified by the *Service Matchmaking*.

We now look at the work that has been done in the area of service interface design. Chiu et al [4] presented a meta-model for service interface as workflow views, which provided a novel approach to derive workflow view from a workflow. By abstracting service interface as a subset of service, it allows internal information to be hidden from external users. However, the work only focused on abstracting a single service interface. In order to support users playing different role in Business Collaboration, Zhao et al [5] proposed the concept of *relative workflow view* by explicitly specifying visibility constraints (Invisible, Traceable,and Contactable) on activities of workflow. Based on different visibility constraint for different users over the same workflow, multiple relative workflow views could be derived for different users that have different relationship with the service (e.g. the retailer service has different relationship with customer and wholesaler).

The idea of applying the concept of *differentiation* in software filed was firstly proposed by Kang et al [6] in the study of Feature-Oriented Domain Analysis which was based on Abstraction and Refinement in a domain. In their work, *Abstraction* represented generic domain products; *Refinement* was used to extend the *Abstraction* to support different domain applications. Cao et al [7] further extended the work in Feature-Oriented Domain Analysis by providing an algorithm to automate abstraction refinement. The idea of service differentiation (DiffServ) was firstly proposed in the area of managing traffic streams in networking applications [8]. For example, certain traffic is treated better than the others in terms of faster handling, more average bandwidth, and lower average loss rate. Veryard [9] argued that the differentiated services should be used as a design pattern in SOC area. He pointed out the need for service differentiation in E-business using an airport example - the airport service needs to meet different requirements of passengers in terms of security, performance and etc. However, no design method is provided to realize the service differentiation. In our previous work [10] [1], a Differentiated Service Design that use policy to control the service to provide differentiated functionalities to users, hence realize service differentiation.

Wombacher et al [2] used finite state machine based model to describe service interface, a matchmaking algorithm was provided based on the model to identify if provided interface is compatible with the user required interface. However, simply showing compatible (or incompatible) is insufficient to help resolve the mismatches. Benatallah et al [11] [12] further classified the compatibility into several categories. Aalst et al [13] define the conformance of service behavior based on fitness and appropriateness. All these work provides a foundation to service adaptation which overcomes the incompatibilities identified.

Based on the matchmaking research work been done, Benatallah et al [14] [15] [3] developed adapter based mechanisms that split/combine messages or swap message sequences to resolve certain types of mismatches. Because the adapter is developed only based on the interfaces, it can not support policy negotiation or process modification to overcome mismatches at policy or process level. Using a totally different approach, we propose Adaptive Service Design (ASD) for service provider. The ASD supports policy negotiation and process modification to overcome the mismatches that can not be resolved by adapters.

3 Adaptive Service Design

Instead of supporting a service with one monolithic business process and single service interface, our design method is to separate the generic tasks that are available to all users from the specific tasks with certain context dependent requirements. The design consists of four components: Abstract Business Process, Policy, Policy Configured Business Process (PCBP) and User Oriented Business Process (UOBP).

Abstract Business Process (ABP) consists of a set of activities and relevant edges linking between activities. There are two types of activities: (1) *concrete*

activity that actually performs general tasks for all users; and (2) *abstract activity* that executes different tasks or even execute different processes depending on *Policy*. Abstract activities are linked to *Policy* which defines how contexts aware tasks (or processes) should be performed.

Policy provides the mappings between usage contexts and tasks (or external business processes). Take online shopping service as an example, in the Check-out process, the "customer profile" as an usage context can be retrieved from the "login" activity. By applying 'provide discount' policy, "VIP customer" will get further 15% discount by invoking the 'offer 15% discount' task. Depending on the usage context values, the policy determines how to plug different tasks (or business processes) into the ABP, and generate *Policy Configured Business Processes (PCBP)* which perform different business functions.

After introducing the basic elements of *ABP* and *Policy* in the previous sections, we can now provide a complete picture of how a 'concrete' business process - *Policy Configured Business Process (PCBP)*, is generated. As discussed before, an *ABP* is a process that contains *abstract activities*, which refer to policy templates. A template consists of a set of mappings between context values and business processes (or tasks in a simple situation). The number of tuples (mappings) in a template determines the number of tasks can be performed for the corresponding *abstract activity*. By replacing all the *abstract activities* in *ABP* with policy specified tasks that come with different context conditions, multiple *PCBPs* can be generated. *PCBPs* support users with different functions, and thus realizes differentiated services based on usage contexts. For example, in Checkout process, the loyal customer and normal customer will be supported by different *Context Configured Business Processes*. We regard the process the generate different PCBPs as Service Differentiation. The details of the Service Differentiation can be found in our previous work [1].

Currently we are focus on constructing a mechanism to derive *User Oriented Business Process (UOBP)* based on PCBP for User Required Interface (URI). Each UOBP corresponds to one URI, it is totally compatible with the specific URI. The mechanism consists of three steps:

1. Generating User Service Interface Execution Paths. In this step a infinite set of execution paths [2] can be derived from the user service interface.
2. Each execution path will be matched by the PCBP. If there is any mismatch, the mismatch would be identified as one of followings: message mismatch, process mismatch, and policy mismatch. Solutions as adapter development, process modification and policy negotiation would be suggested correspondingly to resolve the mismatch. An Adaptation Effort List is hence derived to illustrate the total amount work needs to be done in order to adapt the specific execution paths.
3. An Adaptation Effort Tree can be derived by combining all the Adaptation Effort Lists for the execution paths. The tree illustrate the total work needs to be done to adapt the how service interface. User could also choose only certain branches of the Adaptation Effort Tree to partially adapt the user service interface.

4 Conclusion

Service users or applications can often have the same goal but different interaction requirements. In this paper, we proposed and argued the need for *service adaptation* to serve the purpose mentioned above. We believe the adaptive service should dynamically derive multiple User Oriented Business Processes for different users required interfaces. The design is related with following research areas: service description, service differentiation, service matchmaking and service adaptation.

References

1. Tao, A.T., Yang, J.: Context aware differentiated services development with configurable business processes. In: EDOC, IEEE Computer Society (2007) 241–252
2. Wombacher, A., Mahleko, B., Neuhold, E.J.: Ipsi-pf - a business process matchmaking engine based on annotated finite state automata. Inf. Syst. E-Business Management **3**(2) (2005) 127–150
3. Nezhad, H.R.M., Benatallah, B., Martens, A., Curbera, F., Casati, F.: Semi-automated adaptation of service interactions. In Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J., eds.: WWW, ACM (2007) 993–1002
4. Chiu, D.K.W., Cheung, S.C., Karlapalem, K., Li, Q., Till, S.: Workflow view driven cross-organizational interoperability in a web-service environment. In Bussler, C., Hull, R., McIlraith, S.A., Orlowska, M.E., Pernici, B., Yang, J., eds.: WES. Volume 2512 of Lecture Notes in Computer Science., Springer (2002) 41–56
5. Zhao, X., Liu, C., Yang, Y.: An organisational perspective on collaborative business processes. In van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management. Volume 3649., Springer (2005) 17–31
6. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon University Software Engineering Institute (November 1990)
7. Cao, F., Bryant, B.R., Burt, C.C., Huang, Z., Raje, R.R., Olson, A.M., Auguston, M.: Automating feature-oriented domain analysis. In Al-Ani, B., Arabnia, H.R., Mun, Y., eds.: Software Engineering Research and Practice, CSREA Press (2003) 944–949
8. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: Rfc247: An architecture for differentiated services. Available from: http://rfc.net/rfc2475.html (1998)
9. Veryard, R.: Design pattern: Differentiated service (fewer interfaces than components). CBDI (December 2000)
10. Tao, A.T., Yang, J.: Supporting differentiated services with configurable business processes. In: ICWS, IEEE Computer Society (2007) 1088–1095
11. Benatallah, B., Casati, F., Ponge, J., Toumani, F.: Compatibility and replaceability analysis for timed web service protocols. In Benzaken, V., ed.: 21èmes Journées Bases de Données Avancées, BDA 2005, Saint Malo, 17-20 octobre 2005, Actes (Informal Proceedings)(BDA). (2005)
12. Ponge, J., Benatallah, B., Casati, F., Toumani, F.: Fine-grained compatibility and replaceability analysis of timed web service protocols. In Parent, C., Schewe, K.D., Storey, V.C., Thalheim, B., eds.: Conceptual Modeling - ER 2007, 26th International Conference on Conceptual Modeling, Auckland, New Zealand, November

5-9, 2007, Proceedings (ER). Volume 4801 of Lecture Notes in Computer Science., Springer (2007) 599–614
13. van der Aalst, W.M.P., Dumas, M., Ouyang, C., Rozinat, A., Verbeek, E.: Conformance checking of service behavior. ACM Trans. Internet Techn. **8**(3) (2008)
14. Benatallah, B., Casati, F., Grigori, D., Nezhad, H.R.M., Toumani, F.: Developing adapters for web services integration. In Pastor, O., e Cunha, J.F., eds.: Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005, Proceedings (CAiSE). Volume 3520 of Lecture Notes in Computer Science., Springer (2005) 415–429
15. Kongdenfha, W., Saint-Paul, R., Benatallah, B., Casati, F.: An aspect-oriented framework for service adaptation. In Dan, A., Lamersdorf, W., eds.: ICSOC. Volume 4294 of Lecture Notes in Computer Science., Springer (2006) 15–26

An Architecture Approach to Dependable Trust-based Service Systems

Suronapee Phoomvuthisarn
supervised by Yan Liu

National ICT Australia (NICTA), Australia, & School of Computer Science and Engineering, University of New South Wales, Australia
{suronapee.phoomvuthisarn, jenny.liu}@nicta.com.au

Abstract. *A key challenge in emerging service-oriented computing is the need to establish trust-based loosely coupled partnerships between previously unknown services. A dependable trust framework is essential to capture and maintain realistic trust information based on different requirements of participating services. Most existing trust management frameworks assume that the participating services perform cooperative behavior rather than have strategic behavior for their own interests. In this paper, we propose a novel architecture approach to integrate auction mechanisms into trust framework in order to prevent benefits from untruthful incentives. This is achieved by defining an auction-based trust negotiation protocol and realizing it in the trust framework. This research also aims to examine the impacts that the resulting architecture has on the attributes of dependability in particular trustworthiness, performance and reliability as well as their dependencies. Test cases that capture the key scenarios of these quality attributes are devised and exercised to collect empirical evidence.*

1. Introduction

In emerging service-oriented systems that require exchanging the information with other participants, especially previously unknown services, trust management framework is needed for particular services to allow others to access its resource. To capture and maintain trust information that is distributed over these environments, a dependable trust application is needed for establishing trust to satisfy the security requirements of all the participating services. However, participating services in trust negotiation are not merely to perform cooperative behavior, they also have strategic behavior. For example, some services have their own incentives to reveal their information. In some cases, they might untruthfully reveal their trust information to the service provider, leading to arbitrage opportunities for access to resources. One challenging question is how to prevent the gain from untruthful strategic behavior in order to achieve welfares for all the participants.

Existing trust frameworks have been proposed such as Trust-Serv [1], TrustBuilder [2], and Requirements-Driven Trust Framework [3]. These frameworks mainly focus on

trust negotiation strategies and access control policy. However, existing trust frameworks have not addressed how to prevent strategic behavior. The extension to the existing frameworks should be in place to prevent this strategic behavior from untruthfully revealing their trust information.

Auction mechanism is a suitable technique to solve real-world problems especially in market trading so as to achieve specific resource allocation goals. In this research, we propose an architecture approach integrating VCG (Vickrey-Clarck-Groves) auction mechanism with an existing trust framework [3] to prevent strategic behavior. This also involves defining a new trust negotiation protocol that encapsulates each steps of VCG auction mechanism. Due to the extra computing incurred by VCG, the resulting architecture has an impact on several attributes of dependability, in particular, trustworthiness, performance, and reliability. Moreover, they can have dependencies. For example, when using VCG, extra communication message may impose extra delay to response time and also impact mean time to failure, and therefore the increasing trustworthiness may come at a cost of performance and reliability. In this work we also consider these architectural dependencies and evaluate our solution using an established architecture evaluation method [4].

Our contribution is complementary to existing trust framework with the extra capability to prevent strategic behavior. The use of auction mechanisms induces an effective trust negotiation by preventing a trust-based application from being exploited by incentive services while ensuring accurately trust information captured. This leads to more trustworthy system, as well as more efficient trust negotiation using lightweight mechanisms rather than sophisticated implementations for capturing trust information. Also the novel architecture will be optimized to support performance and reliability.

The rest of the paper is organized as follows. We start with related work in Section 2. The motivating scenario is described in Section 3. The research method is presented in Section 4. We then propose the architecture and negotiation protocol in Section 5. Evaluation plan is discussed in Section 6. This paper concludes at Section 7.

2. Related Work

Existing trust frameworks have mainly focused on trust negotiation strategies and access control policy. Key among existing related work includes Trust-Serv [1], TrustBuilder [2], and Requirements-Driven Trust Framework [3]. Trust-Serv, a model-driven trust framework, uses state machines to represent and determine credential exchanges for access to resources [1]. TrustBuilder uses credential disclosure trees and negotiation strategies to facilitate protection of credential information during negotiation [2]. Finally, Requirements-Driven Trust Framework, our existing work, combines trust negotiation and trust level computation together based on the service requirement [3]. This framework is very suitable for a distributed environment where trust is negotiated based on the service requirements of each domain involved. However, existing trust frameworks have not

addressed how to prevent strategic behavior. In this research, we propose a novel architecture approach integrating VCG (Vickrey-Clarck-Groves) auction mechanism with requirements-driven trust framework [3] to prevent strategic behavior.

3. Motivating Scenario

In this section, we provide a simple scenario to explain how VCG auction mechanism can be used to prevent strategic behavior of services. We postulate an application where several travel agents are competing for deals to manage group travels. A customer as a traveling group can register the request with a travel agent brokering service. Travel agents compute with each other by using their web services to provide its quote to the broker with its credit. Hence, an appropriate identity should be provided to the broker to check for credits. The higher the submitted credit, the higher sensitive credentials needed to prove their identities.

Suppose the particular travel agent untruthfully reveals its credit to the broker, this agent would get the deal. In this case, the system is untrustworthy because of its lacking capabilities to prevent this strategic behavior. One solution is to implement VCG in trust management framework that prevents any gain from untruthfully revealed information. In VCG, the bidder who submits the highest bid wins the auction and pays the second-highest bid [5]. This principle of VCG is that "lying does not pay", which means bidding something other than the bidder's true value is never beneficial and sometimes was detrimental with penalty. Suppose all travel agents truthfully send their credits to the broker. Let's say, agent A, B, C, D submits 10, 20, 50, and 60 credits with the same quote, respectively. In this case, D has been chosen to get the deal and is required to prove the credentials based on 50 credits. The agent D will have the net utility gain of 60 minus 50 which is 10 credits. Note that the credentials to be exposed for 50 credits would be less trustworthy than 60 credits. If agent D service sends its credit more than 50, the result remains the same. If agent D sends its credit less than 50, it loses the competition. If other travel agents such as agent B send its credits higher than 60, suppose 80 credits, it would get the deals but have to pay the net utility gain of 20 minus 80 which is negative and eventually a loss. Therefore, all travel agents are content to send their truth credit.

4. Research Method

In this research, an architecture approach is proposed to develop the solutions for dependable trust-based service oriented applications. A negotiation protocol encapsulates VCG auction mechanism is defined and realized in relevant VCG mechanism components. These components interact with trust components in an architecture framework. It is also aimed to optimize this architecture with regards to the impact on the attributes of dependability which are trustworthiness, performance, and reliability. The

improvement solution will be based on observing the dependencies among these qualities of attribute using test cases. Further architecture improvement will be devised given the empirical evidence. The proposed research methods consist of three stages as follows:

4.1 An architecture approach is conducted at the architecture level to develop the novel architecture for dependable trust-based service oriented application. The essence of this stage is to build a reference trust framework architecture integrating with auction mechanisms, in particular, VCG mechanism to prevent strategic behavior. This is achieved by developing negotiation protocol encapsulating with VCG and realizing it in relevant VCG components. The primary concern with the resulting architecture is to ensure the separation of concerns between trust and VCG components. Trust and VCG components have to overlap in functionality as little as possible so that trust-based application can be maintainable and extensible with other auction mechanisms. Although the resulting architecture can help preventing strategic behavior, there is a possibility that a poorly-designed architecture might degrade the composition capability of the original trust framework. Loosely-coupled component-based approach can be used to decompose system into functional components to support extensibility and maintainability. Our plan is to develop a basic trust management architecture as a prototype with the motivation scenarios deployed. Then, the auction mechanism is integrated into the trust negotiation protocol and interacts with the trust management framework. It should be tested that the VCG mechanism is efficient for preventing strategic behavior.

4.2 The core architecture is extended with performance and reliability qualities of attribute. The essence of this stage is to examine the dependencies of performance, reliability, and trustworthiness based on the impacts that resulting architecture has on. One such challenge is to address the dependencies of these attributes of dependability which may incurs trade-off in the architecture design and implementation. For example, increasing trustworthy trust-based application with auction mechanisms might decrease system performance in response time and increase points of failure that affect reliability. Our solution is to use the basic prototype as a test-bed to further identify and pinpoint any architecture issues incurred by the computation of the auction mechanism. Test cases are devised to observe and measure the performance and reliability of this architecture. Moreover, the dependencies or even correlations between trustworthy, performance and reliability are studied. The root cause is then resolved either at architecture design or at the implementation level.

4.3 Evaluation is conducted at the empirical level to firstly evaluate appropriateness auction mechanisms of the extended architecture used for preventing strategic behavior. We then evaluate the value of other dependability attributes, performance and reliability. One challenge is that trustworthiness is a qualitative attribute. It is quite subjective to evaluate the trustworthiness of trust framework. MEMS [4] has proposed the scenario-based architecture evaluation method used to evaluate both quantitative and qualitative attributes, hence, we will adapt MEMS to justify the value of these attributes. The plan is to devise three case studies with each having the emphasis on trustworthiness, performance and reliability. All case studies are to be

modeled after real-world scenarios in service oriented architectures. At the end, architecture design and implementation guidelines are provided with regards to different characteristics of service oriented applications.
Section 5 discusses the current progress in the first step.

5. The Architecture

A conceptual trust management architecture is proposed to prevent strategic behavior. This section discusses the current progress in the first step of the method. We also exemplify this architecture using the auction-based trust negotiation scenario.

Trust-based application has to be extensible while negotiation protocol has to be maintainable when changing in auction mechanisms. This is achieved by using loosely-coupled component-based approach that decomposes system into functional components with well-defined interface used for communication between components. The key components of this architecture are Trust Engine and VCG Engine.

Trust Engine is responsible for trust negotiation between services. This engine involves establishing the services that will be exchanged between the participating services and establishing a negotiated trust level for service access. It consists of the following components:

- Trust Negotiation Module is responsible for checking the validity of *trust tokens*, a set of selected credential(s), based on specified *trust token type*, a set of attributes and the range of values they should be constrained to [3]. If all attributes in the certificates satisfy all *trust token types*, they are proved to be valid. For example, the broker's *trust token types* based on 50 credits are ({firm's age > 10}, {location = U.S.A.}). In the context of this example, travel agent B has to provide the credential(s) stating that B's company has been established for over 10 years and its location is in USA.
- Trust Level Computation Module is responsible for computing trust level that service provider has on service requester.

Auction Engine is responsible for computing the appropriate trust level that service provider has on service requester based on auction mechanisms calculation. It consists of the following components:

- VCG Mechanism Computation Module is responsible for computing trust level based on VCG auction mechanism that can be categorized into single-item and multi-item.
- Utility Calculation is responsible for calculating the net utility gain for services.

As a result, in this architecture Trust Level Computation Module uses the VCG Mechanism Computation Module to compute the appropriate trust level of particular resources service requester requests. The service that sends the highest credit will be required to send the credential(s) based on the second-highest credit submitted. In this case, credit is the trust information that will be translated into trust value. The higher the credit, the higher the trust value will be. After each VCG steps, the selected service then continues to exchange the certificates based on each *trust token types* required for trust

level of particular resources. The higher the trust level, the higher sensitive credentials needed to prove their identity.

6. Evaluation Plan

The basic architecture has been implemented with a set of travel agent web services using Apache Axis1.0. Trust negotiation protocol is implemented using JXTA technology which includes a set of open peer-to-peer protocols. In this earlier stage, to demonstrate the trust management framework capabilities when using VCG auction mechanism, the plan is to implement two prototypes with and without VCG deployed. The testing is presented as follows:

- Utility test evaluates the practical usage of the architecture in the case that it can efficiently avoid untruthfully strategic behavior.
- Overhead test evaluates the performance in terms of response time of the architecture when using VCG.

7. Conclusion

In this early stage, we envision a conceptual architecture of trust-based service oriented application integrating with VCG auction mechanism. Our architecture introduces the solutions of dependable trust-based application that can prevent strategic behavior. The notion of VCG induces an effective trust negotiation by preventing a trust-based application from being exploited by incentive services.

References

1. Skogsrud, H., Benatallah, B., Casati, F., “Model Driven Trust Negotiation for Web Services”, IEEE Internet Computing, November-December 2003, Pages 45-52.
2. Yu, T., Winslett, M., Seamons, K. E., “Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation”, ACM Transactions in Information Systems Security, Vol. 6, No.1, February 2003, Pages 1-42.
3. Phoomvuthisarn, S., “Trust and Role Based Access Control for Secure Interoperation (“TracSI”)”, Communications and Information Technologies, 2007, 17-19 Oct. 2007 Page(s):1458 - 1463
4. Liu, Y., Gorton, I., Bass, L, Hoang, C., Abanmi, S., MEMS: A Method for Evaluating Middleware Architectures, QoSA, June 27-29, 2006, Sweden. Lecture Notes in Computer Science, Volume 4214/2006, 2006.
5. Ye, S., Makedon, F., Ford, J., “VCG computational Mechanism”, in Proceedings of the 4th International Conference on Peer-to-Peer Computing, 2004. 25 Aug. 2004 Page(s):108 – 115

Authorization Control in Business Collaboration

Daisy Daiqin He
Supervised by Jian Yang

Department of Computing, Macquarie University,
North Ryde, NSW 2109 Australia
daiqin,jian@ics.mq.edu.au

Abstract. Authorization control has been well studied for years, and there are quite a few theories and techniques available for handling access control for a single or a centralized system. However unique and challenging security issues concerning business collaboration in the context of service oriented computing (SOC) have arisen due to the dynamic and loosely coupling nature of the environment in which business collaboration is conducted. In this paper, we discuss different authorization control issues in business collaboration and present an overview to our proposed PD-AC framework, which we believe it has laid a good foundation for future work in the area of policy consistency checking, policy negotiation, and security policy enforcement in business collaboration.

Key words: access control, security policy integration, collaboration

1 Introduction

Web services and Service Oriented Computing (SOC) provides infrastructural support for cross-organization collaboration in distributed environments. However security concerns become one of the main barriers that prevent widespread adoption of this new technology. Each organization or business unit has its own interest and security polices for defining who has access right for specific services and how services can be used. In web services environment with complex cross-organization collaborations, Different security challenges will arise with different number of participants involved in collaboration.

Access control is enforced in a single organization by using pre-defined authorization control polices. Common authorization control practices include requester credential verifications, role assignments and access decision makings.

In collaborative business world, a service can be accessed by a party which can pass it to other parties. We shall use some examples in health care to illustrate some issues.

Suppose a patient granted access right to a General Practitioner (GP) in a medical centre on patient's health record. Since the physician is a member of a research institute, he could also let researchers in this institute access to this health record based on the security policy of the medical centre. However the patient may not want the GP give access right to anyone on his health record

unless there is an emergency. How can we use security policies to control the way in which information or service is propagated between organizations?

Problem can also arise from service composition. For example, a medical center allows the patients who hold an OSHC (overseas student health cover) to book appointments on line for general inquiry and ultrasound exam based on its policy. A radiology institute wants to collaborate with the medical center and accepts on line bookings from the medical center. However, OSHC is not accepted by the radiology institute's policy. It is challenge to decide which authorization control policy they should follow if these two organizations collaborated in the presence of this policy conflict.

Moreover, organizations collaborate with each other in various ways. Before organizations engage in collaboration, their authorization policies need to be analyzed to decide the possibility of collaboration under the authorization constraints defined by each individual party. Therefore, we need to evaluate consistencies of access policies of different organizations. Intuitively, the concept of 'access policy consistency' is referred as that the access policies of different organizations are conflict free, for the same service. And organizations are able to collaborate in the intended way securely in terms of access control policies.

To address these complex security policy issues, we need a framework that can analyze, evaluate and integrate security policies if necessary for collaboration purpose, based on which negotiation can be guided and security integrity can be enforced. Security control in business collaboration should take individual organization's access policy into account, as well as the type of the collaboration which is referred as collaboration pattern in the paper.

Access control issues in signle organization or signle domain have been well studied [5, 3, 2]. Access control in collaborative environment has just started to attract the attention of the research community [1], but little attention have been given to consistency study between access control policies of different collaboration participants, particularly in the context of Web Services. Furthermore, these studies only focused on providing solutions to some aspects of security issues in terms of: security policy specification, access control in distributed environment, and access decision making. What is missing is a comprehensive analysis of: (1) what security requirements really are in the context of business collaboration; (2) security policies can be specified; (3) how security policy can be verified, evaluated, and integrated for the purpose of business collaboration. No feasible mechanisms can be developed for policy negotiation and enforcement without this analysis.

The rest of paper is organized as follows. We first explain the basic elements included in the security policy and relationship between these elements in Section 2. In Section 3 we identify and model different types of collaborations. In Section 4 we model and discuss security controls for different types of collaboration in health care environment. We propose our framework in Section 5. Formal definition and of policies and rules are presented in Section 6. Related work is discussed in Section 7. In Section 8, we give some concluding remarks and outline future research directions.

2 Related Work

A number of studies concentrated on authorization architecture [7, 4]. Author in [7] suggested a brokered architecture to build composite Web services according to the specified security constraints. They used security matchmaker to find right collaboration partners who have compatible security policies, which is similar to our research. However, it did not address the issue that inconsistencies and conflicts exist between security policies of prospective partner.

A few of studies have touched policy level security issues, which focused on identifying different security requirements and proposed specifications for these requirement. Trust-Serv[11] modeled access control processes in web services using state machine and provides lifecycle management for policies. Ws-AC [6] provides an adaptive system that is capable of asking users to refine their requests to comply with security policies. Again, all of them are concerned security policies in a single organization and none of them addressed policy problems in collaborative environment.

There are few papers on Web service authorization control in the collaborative environment. We are aware of the work presented by [1], which presented a framework for managing authorization policies for Web service compositions. [12] proposed an approach to security policy integration and conflict reconciliation. But they neglected the fact that different types of collaboration affect the way the collaboration policy is developed as well as the requirements on collaborative partner's authorization policy. An evaluation on collaborative partner's access policy has to be carried out before the collaboration be established. Our work is to fill in this gap. We believe this is the first step toward conflicts detection and suitable collaboration partners discovery.

In summary, none of these studies went deep into different types of cross-organization collaboration, which could raise different requirements on access control policy of prospective collaborative partners. Our goal is thus to provide a framework that could identify types of cross-organization collaborations in this context; define collaboration requirements in terms of security policy; generate policy integration rules. Most importantly our work is in the context of business collaboration which involves multiple organizations rather than simple interaction between individual requester and single service provider.

In this paper we present some of our initial results in modeling security requirements and integrating security policies for business collaboration. We believe this study is the first step towards achieving an understanding of a secured of business collaboration in terms of authorization control.

3 The PD-AC Framework

The core function of our PD-AC (Policy Driven Authorization Control) framework is realized by a Policy Evaluation Engine, which is associated with every organization that provides services in business collaboration, see Figure 1. Policy Evaluation Engine is used to analyze the nature of collaboration, make access

decision, and finally generate the collaborative security policies. The proposed framework consists of two components: request mediator and policy evaluation engine. Upon receiving a request for a service, the request mediator firstly identifies the type of requested service, which is a process that identify whether the requested service is a collaborative service. If it is a simple service that does not involve any collaboration, the mediator will perform normal access control functions: looking up database, check which role in the database has the requested privilege; compare requester's credential with security requirement; make access decision.

However, if it is a collaboration request (prospective collaboration partner), the request mediator will pass the request to the Policy Evaluation Engine to perform the following functions:

- Identify requested collaboration type;
- Evaluate requester's authorizaton policy according to policy requirements for requested collaboration type;
- Make collaboration decision;
- Generate collaborative authorization policies for the collaborated service if the requester if acceptable (from previous step);

In the following subsections, we will discuss different types of collaboration and policy evaluation engine in more detail.

3.1 Business Collaboration Patterns

Business collaborations consist of complex relationships and interactions among organizations. Authorization policies of all participate organizations need be carefully considered and evaluated. Our analyze shows that different collaboration types affects the requirements on collaborative partner's authorization policy. We have conclude four different ways of collaboration between organizations and provided simple examples in Health Care domain [8].

- **Simple Access** (**SA**): it depicts the most simplest 'request service - provide service' scenario that involves two organizations.
- **Composite Services**. The Composite Service we discuss here is referring to the service that is based on the integration of multi-service providers. Two different cases are identified in service composition:
  1. **Composite service with agent** (**CSWA**): Multiple numbers of service providers provide their services through an centralized agent, i.e. health insurance company and health service providers.
  2. **Joined service without an agent** (**JSOA**): Two organizations involving in a peer-to-peer collaboration and provide a joined service by integrating their business processes or integrating part of their business processes together to form a new service directly without any agent.
- **Service Outsourcing** (**SO**): As the result of globalization, Outsourcing and offshore Outsourcing has become a popular trend in many industries, SO depicts collaboration relationship between outsourcer and outsourcees.

- **Service Propagation** (**SP**): it depicts collaborations that involving multiple organizations and 'forward' privilege could be passed from one organization to another organization.

3.2 Policy Evluation Engine

Authorization policy of prospective partners are compared and evaluated in Policy Evaluation Engine to determine the suitability for requested collaboration pattern. Before we can compare and evaluate policies from different organizations, we need to understand all the necessary elements and their relationships for a generic authorization policy. Therefore, an authorization policy model is proposed to specify authorization policy in an individual organization. We base our policy model on Role-Based Access Control (RBAC)[9] and encoded the model in Description Logic. The main entities in the model are roles, credentials, privileges, obligations and provisions[8]. Policies from two different organizations can be compared by combining them into a single model.

Three categorizes of inconsistencies have been discussed in our work: role, credential and privilege inconsistencies[10]. Each category consists of several inconsistency types. We use a Description Logic reasoner (an automated proof engine) to analyse the inconsistencies in policies. We encode the inconsistency tests as concepts and relations in our model. Individual policies expressed using the model can then be combined and tested. Given a combined policy, with the roles and privileges of the two organisations suitably related, a reasoner will prove that the tests are either satisfiable or unsatisfiable and these results can be analysed to check if they satisfy the requirements for the particular collaboration. Since the tests are part of the general model, so they can be expressed once, proven to encode the required meaning and used to testing any two policies.

We have identified several cross-organization collaboration patterns, different collaboration pattern can result in different requirements for authorization policies of prospect collaborative partners. The requirements we discussed in our work are basic requirements that must be satisfied by the prospect partner to be considered for requested collaboration. Policy requirements for different collaboration patterns are analyzed. Inconsistencies are discussed for each collaboration pattern[10]. Depends on the collaboration pattern, some of the inconsistencies are acceptable, some of them needs further negotiation and some of inconsistencies lead to reject.

4 Conclusion

In this paper, we proposed an authorization conrol framework for business collaboration. Our analyze shows that different ways of collaboration could affects the requirements on authorization policy of collaborative partner. The proposed framework use a policy evaluation engine to analyze collaboration suitability of prospective partners for requested collaboration pattern from authorization policy perspective. In our previous work, we have concluded different business

collaboration patterns and discussed different requirements for the prospective partner's authorization policies in the collaboration. An description logic based authorization policy model has been proposed to specify the authorization policy of an individual organization. Inconsistencies between authorization policies from different collaboration participants are identified and classified based on the model, based on which the collaboration possibility are analyzed. In the future we intend to extend this work to incorporate the following:

- Context constraints that affect access control.
- Inconsistencies that caused by role hierarchies and separation of duty.
- Collaboration access control policy requriements from business transaction and process perspective.

References

1. Rouached, M., Godart, C.: Reasoning about Events to Specify Authorization Policies for Web Services Composition. In: 2007 International Conference on Web Services, IEEE Press, Salt Lake City (2007)
2. Srivatsa, M., Iyengar, A., Mikalsen, T., Rouvellou, I., Yin, J.: An access control system for web service compositions. In: 2007 International Conference on Web Services, IEEE Press, Salt Lake City (2007)
3. Kagal, L., Paolucci, M., Srinivasan, N., Sycara, K., Denker, G.: Authorization and Privacy for Semantic Web Services. IEEE Intelligent Systems. 19, 50-56 (2004)
4. Ziebermayr, T., Probst, S.: Web Service Authorization Framework. In: 2007 IEEE International Conference on Web Services, pp. 614-621. IEEE press, San Diego (2004)
5. Sirer, E. G., Wang, K.: An access control language for web services. In: 2002 SACMAT, pp.23-30. (2002)
6. Bertino, E., Squicciarini, A. C., Mevi, D.: A Fine-Grained Access Control Model for Web Services. In: IEEE International Conference on Services Computing, pp. 33-40. IEEE press, Shanghai (2004)
7. Carminati, B., Ferrari, E., Hung, P. C. K.: Security Conscious Web Service Composition. In: IEEE International Conference on Web Services, pp. 489-496. IEEE press, Chicago (2006)
8. He, D. D., Yang, J.: Security Policy Specification and Integration in Business Collaboration. In: 2007 IEEE International Conference on Services Computing (SCC 2007), pp. 20-27. IEEE press, Salt Lake City (2007)
9. Sandhu R. S., et al.: Role-Based Access Control Models. IEEE Computer. 29, 38-47 (1996)
10. He, D. D., Yang, J.: Identify Authorization Control Requirement in Business Collaboration. In: IEEE Service Oriented COmputing (2) 2008 (SCC 2008). IEEE, 2008
11. Skogsrud, H., Benatallah, B., Casati, F.: Trust-Serv: Model-Driven Lifecycle Management of Trust Negotiation Policies for Web Services. In: 13th World Wide Web Conf (WWW 2004). ACM Press, New York (2004)
12. Yau, S. S., Chen, Z: Security Policy Integration and Conflict Reconciliation for Collaborations among Organizations in Ubiquitous Computing Environments. In: UIC'08, pp. 3-19. (2008)

TPIM: Transparent Privacy-Enhanced Identity Management of Web Services

Yong Yang[1,2*]

[1]School of Computer Science, University of Electronic Sci. & Tech. of China, China
[2]Department of Computing, Macquarie University, Australia
yongyang@ics.mq.edu.au

Abstract. The growth of web services has been accompanied by sharing more and more users' personal information with service providers, which has raised concern about possible malicious or accidental unauthorized abuse of user information. This paper focuses on how we can give the user a deep sense of safety, privacy and certainty about service invocations in the diverse and heterogeneous computing environment. We present *Transparent privacy-enhanced Identity Management of Web Services* (TPIM), a privacy-enhanced personal Identity Management architecture for web services users. TPIM is an extension of SOAP specification, which provides a sense of "circle of trust" in the identity management during the collaborations of web services. It enables that user's identity or personal data to adapt to be accessible only to whom they trust. In other words, a user can put his or her personal information on any web services and maintain privacy in different user-defined security level (including up to unconditional anonymity) as well.

1 Introduction

People are expected to remember different organization-specific user names and passwords in the online world. Identity management systems seek automated solutions for managing their identities by making them transferable across organizational boundaries. However, an increasing sharing personal information with service providers concerns the user with risks to privacy. Aside from the end-users' privacy, if the system is perceived as privacy infringing, it will endanger the reputation of involved service providers, which may lead to loss of profits in the long run.

Research has shown that how to manage the identities in web services and maintain user's privacy is really a challenge. Many efforts are made at "domain-centric" identity management, in which users have no control, and suffer from the identity theft or fraud. So scientists shift focus onto the dimensions of users control, where there is no universal agreement to date.

* This work was performed during the author's scientific visit at Department of Computing, Macquarie University, Australia.

In this paper we investigate a *transparent privacy-enhanced Identity Management* (*TPIM*), which enables the users have total control over the management of their identities. In order to enhance users' privacy, the SOAP standard is extended and a TPIM framework supporting "Single sign-on"(SSO) is proposed, which allows the user to access multiple sets of resources after being authenticated just once. It provides users with a more seamless user-experience when accessing different user accounts on the Internet.

To sum up, this paper makes the following main contributions:

- Id-based Ring signature is introduced and adapted to support unconditional anonymity. Even if ID information is leaked later on, the user can not be identified. Meanwhile the control of privacy preserving shifts from the third party to users themselves, which greatly increases users' confidence and promotes privacy.
- The SOAP architecture is extended to enhance privacy in web services. The user can manage her own profiles and have a total control on her identities. The user can set different levels of security identity. For example, a user may use a set of credentials or id name to access her blog with security level 1, a second set to discuss work with her colleagues with security level 2, a third set to purchase goods online with security level 3. Besides, a novel rule model is presented to exploit the privacy policies on both the organizational and execution levels.

2 RELATED WORK

Privacy in general has been exploited for years. However, privacy in web services is still under development. Research to date has been focused on developing privacy languages. Rezgui et al. [5] investigate the feasibility and provable reliability of privacy preserving solutions for web service infrastructures. Yee [9] and Ni et al. [3, 4] designs privacy controllers together with user privacy policies to protect privacy. Squicciiarini et al. [8] provide a set of assertions to define the privacy related properties. But none of them addresses the issue of enforcing privacy that confirms to emerging industry standards. Most commercial available systems such as Microsoft .Net Passport and Liberty Alliance can be improved on the user-friendly feature. Without consideration of unconditional anonymity, [2] presents a personal Identity Management, which can be a complement with privacy enhancement.

In cryptography, Sharmir [7] introduced the notion of identity-based (ID-based) cryptography to solve the certificate management problem, which is supposed to provide a more convenient alternative to the traditional public key infrastructure (PKI). Ring signature [6] is a type of digital signature that can be performed by any member of a group of users that each has keys. But it can not be determined which of the group members' keys was used to produce the signature. The combination of ID-based cryptography and ring signature schemes has been well-studied in the recent research. Chow et al. [1] proposed a high efficient construction of ID-based ring signature, which only needs two pairing computations for any group size.

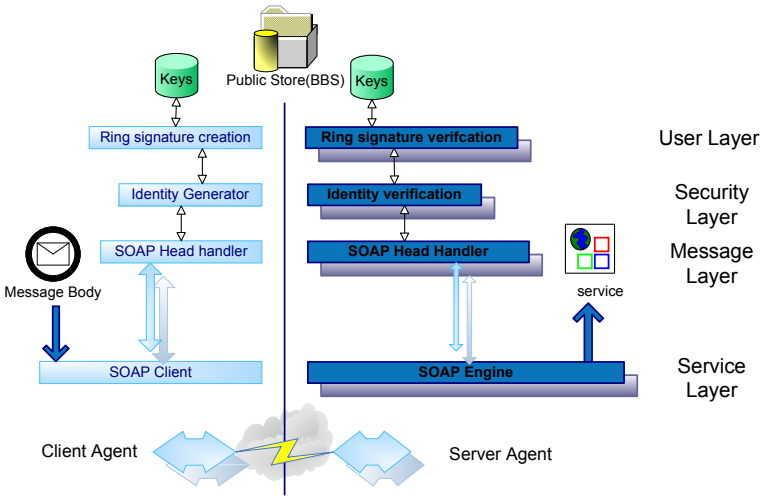

Fig. 1. Privacy-enhanced framework

3 Architecture of TPIM

3.1 Design and usage scenario

The general idea of unconditional anonymity in TPIM is to hide the user's identity in a group S during service invocations. Figure 1 illustrates the architecture for our TPIM framework. In order to be convenient for leveraging applications software, our framework does not break any existing services by acting as add-on components, which guarantees easy integration with existing web-based applications. Specifically, TPIM agents will probe in the network layer and snatch SOAP packages during the monitoring. Once identity related packages are intercepted, they are forwarded to user space to reconstitute the conversation for further judgement. After identity verification, the packages are either dropped or injected back to network layer. All the procedure are well encapsulated and executed in the background, making it completely **transparent** to the end-user.

We extend SOAP specification to support security and privacy features discussed in this paper. The <wsse:security> head blocks are designed to carry privacy related attributes:

- **ValueType**: A string identification label defines the value space and type of the encoded binary data. The value we have chosen for our anonymous group identification security token is "IdBasedRingSignature".
- **EncodingType**: It defines the encoding format of the binary data. In our protocol it is set to "wsse:Base64Binary" to denote a base64 encoding.
- **NameID**: This element describes the group S which the user choose to hide in. To promote privacy, make sure the members within their lifespan during the period of invoking. We can use colon (:) marks to concatenate all the identifiers of individuals in the group S. For instance, if such group includes three persons: Alice, Bob and Lily, the NameID should be "Alice:Bob:Lily".

– **Conditions**: Conditions must be evaluated when assessing the validity of the assertion. *NotBefore* and *NotOnOrAfter*, together with *IssueInstant* define the exact lifetime of the assertion.
– **AttributeSatement**: It asserts a multi-valued attribute associated with the authenticated principal. In the response assertion, all the group public keys information is linked by colon (:) with each other in the same order of NameID element. For instance, the attribute values for Alice, Bob and Lily may be "XD6s. . . :ZCCA. . . :ors. . . ". In addition, the correspondent life expectancy is further supplied to assure the validity of each individual.

An example of a SOAP header containing anonymous group identification is presented in Figure 2. This extension gives rise to an additional payload required for encoding anonymous identification tokens in SOAP request that is proportional to the size of the group the user belongs to.

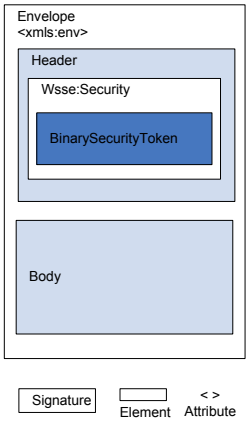

Fig. 2. SOAP extension

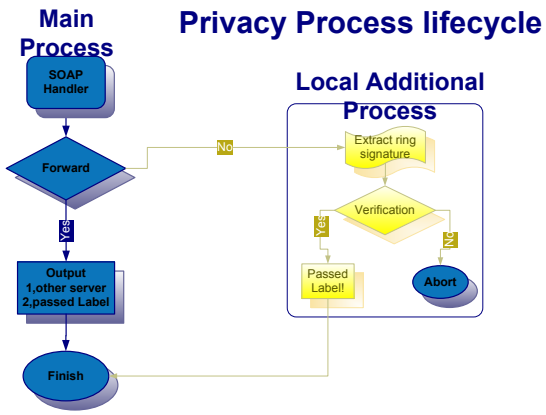

Fig. 3. Privacy enhanced structure

3.2 Privacy enhanced process

During the invocation, when the user issues a SOAP request toward Web services, the message is implicitly intercepted and processed by the client agent. This handler invokes the identity generator module and prepends the resulting identification token together with a timestamp to the SOAP header blocks of the outgoing request. The identity generator will comply with user's directive and bind the request to corresponding identity profile. For example, in the highest security user profiles, the Id-based ring signature is produced to attain unconditional anonymity.

Whenever the service provider receives a SOAP request from client agent, the server side agent is implicitly invoked to determine whether the request should be accepted or not. If the request is for an authorized Web Services and no group-relevant identification information are provided then it is rejected by raising a *SecurityTokenUnavailable* SOAP fault. In the case that the timestamp reported

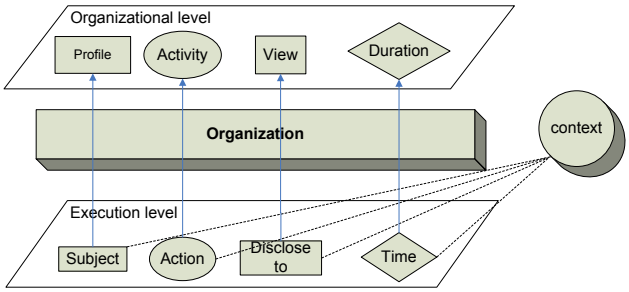

Fig. 4. rule model for TPIM

in the request is older than a fixed security time interval the request is rejected with *FailedAuthentication* SOAP fault. Otherwise, the identification request is processed by identity verification module. If the verification is successful then the service request is executed and the response is returned to the application client. Otherwise, a *FailedAuthentication* SOAP fault is sent back to the requesting client. A representation of the privacy process life-cycle is depicted in the Figure 3. In order to avoid the flow peak in SOAP header request, we forward privacy process to other available server agents for load-balancing.

3.3 A TPIM rule model

As shown in Figure 4, a rule model is designed to facilitate user's privacy policy setting under web service circumstances. Each security policy is defined for and by an organization. Thus, the specification of the security policy is completely parameterized by the organization so that it is possible to handle simultaneously several security policies associated with different organizations. The model is not restricted to identity permissions, but also includes the possibility to specify other identity related information such as priorities.

The rules are context sensitive, so the policy could be expressed dynamically at two different levels.

1. *Organizational level*: The users defines privacy rules through abstract entities (profile, activity, view, duration) without worrying about how each organization implements these entities.
2. *Execution level*: When a user login in other organization, the execution authorizations are granted (or not) to him according to the execution rules. TPIM maps from organizational level to execution level for further elaborate control.

The derivation of invocation policies can be formalized as : Rule $\Gamma = Permission \times \Upsilon \times H$ while $Permission(s, \alpha, d, t, c)$ is defined as $\forall$ subject $s \in S$, performs action $\alpha \in A$, login on to disclose-to service $d \in V$, at time $t \in D$.

- Profiles S: A set of identity profiles in different security levels.
- Activity A: A set of aims of identity requests.

- View V: a set of other services whom the identity information can be disclosed to.
- Duration D: A set of durations of validity with regard to identity information.
- Privacy level Υ: The identity information should be protected at different privacy level such as whether it allows service providers to store user's identity information.
- Handling H: Once the identity information is breached, what approaches should be issued to notify the user of the risk, such as sending an email or an alert. The event-based approach is well suited for services' distributed environments. Apart from the regular infrastructure, the design will facilitate measures to integrate accounting and notification support.

4 Conclusion

We have introduced Id-based ring signature into web services and extended the SOAP standard to achieve privacy enhancement. The user can have sufficient control on her privacy. It provides a more user-friendly and efficient ways of managing digital identities and enables people to assert their privacy rights in the online world. As future work, we will develop a tool to simulate the rule model and perform conflict detection to help the designer to refine rules.

References

1. S. S. M. Chow, S.-M. Yiu, and L. C. K. Hui. Efficient identity based ring signature. In J. Ioannidis, A. D. Keromytis, and M. Yung, editors, *ACNS*, volume 3531 of *Lecture Notes in Computer Science*, pages 499–512, 2005.
2. T. M. Eap, M. Hatala, and D. Gasevic. Enabling user control with personal identity management. *scc*, 0:60–67, 2007.
3. Q. Ni, D. Lin, E. Bertino, and J. Lobo. Conditional privacy-aware role based access control. In *ESORICS '07: Proceedings of the 12th European Symposium On Research In Computer Security*, pages 72–89. Springer, 2007.
4. Q. Ni, A. Trombetta, E. Bertino, and J. Lobo. Privacy-aware role based access control. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 41–50, New York, NY, USA, 2007. ACM Press.
5. A. Rezgui, M. Ouzzani, A. Bouguettaya, and B. Medjahed. Preserving privacy in web services. In *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*, pages 56–62, New York, NY, USA, 2002. ACM.
6. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.
7. A. Shamir. Identity-based cryptosystems and signature schemes. *Proceedings of CRYPTO*, 84, 1984.
8. A. C. Squicciarini, A. A. Hintoglu, E. Bertino, and Y. Saygin. A privacy preserving assertion based policy language for federation systems. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 51–60, New York, NY, USA, 2007. ACM.
9. G. O. M. Yee. A privacy controller approach for privacy protection in web services. In *SWS '07: Proceedings of the 2007 ACM workshop on Secure web services*, pages 44–51, New York, NY, USA, 2007. ACM.

Measuring similarity of service interfaces

Ali Aït-Bachir*

Supervised by Pr. Marie-Christine Fauvet
University of Grenoble, LIG (MRIM)
385 rue de la bibliotheque – B.P. 53
38041 Grenoble Cedex 9, France

Abstract. In this paper, we present a similarity measure between behavioural interfaces of Web services. This measure computes the difference value of simulation between two service interfaces. In our previous work we implemented an algorithm to detect the exact location of differences between service interfaces in a tool namely BESERIAL [1]. The similarity measure is based on the results of the detection algorithm. In our case study, this measure is used to select the most suitable service to substitute a previous one, which is no longer available at design time.

1 Introduction

Web service interfaces can be described from two aspects: i) The structural aspect models the provided operations, and the schema of the messages that the service can send and receive. These operations can be described by using WSDL for instance. ii) The behavioural aspect refers to the control flow between the operations and establishes their inter-dependencies. In conversational services, such behavioural interfaces can be described using BPEL for instance. Nevertheless, Finite State Machines (FSM) is the formal model adapted in our work to describe behavioural interfaces [7]. In this paper, we do not consider semantic aspects of operation definitions.

Client applications are meant to consume provided operations in a service interface. Conversations between a client application and a service are loosly coupled. Thus, if the service evolves and provides a new interface, then incompatibilities may arise as a client application does no longer match the new interface. The provided interface of a service evolves from a previous definition to a new one by means of basic differences (*addition, deletion* and *modification* of operations). The exact location of these differences can be detected and resolved instead of programing a new client application whose required interface is compatible with the new interface definition. However, if the service is no more available, there is no choice left to developers than to substitute this service by another one, at design time. If there exists no service whose interface simulates the old service interface, it is interesting to discover another service whose interface has a minimum number of differences with the previous one.

* This author is partially funded by the Web Intelligence project granted by the French Rhône-Alpes Region

This paper is structured as follows. First, Section 2 introduces the running example. Section 3 gives details on the quantitative simulation measure. Section 4 shows some experimental results. Then, Section 5 gives a panel of the related work on the diagnosis of differences in service interfaces. Finally, Section 6 concludes and sketches the future work.

2 Case study

As a running example, we consider a scenario where a car factory interacts with one of its provider of goods and services. The service provider describes its operations in WSDL and the control flow is established using BPEL process protocol. Figure 1 (a) illustrates the activity diagram of the provided interface of the provider service. This provider processes service and goods orders from the car factory. The provider receives a service order which can be updated by its client (the car factory) only if the invoice is not sent yet (see the flow which loops back to the *ReceiveServiceOrder* activity). Once the service invoice is sent, the provider waits for the transfer from the client to finally send him the *ShipmentTrackingNumber* (STN). On the other hand, when a *GoodsOrder* is received, a *GoodsInvoice* is immediately sent to the client. This former can either send his *CreditCardDetails*, to pay the invoice, or update his order by sending a new *GoodsOrder* (see the flow which loops back to the *ReceiveGoodsOrder* activity). The client pays the invoice, and then the provider sends him the *STN*.

If the service provider is no more available, the car factory will send an invitation to tender to substitute the old provider and all candidates will provide their behavioural interfaces. The selection criterion is that the provided interface of the new partner must conform as much as possible to the required interface by the car company. In other words, the new provider is such as there exists a minimum number of changes in the new provided interface in order to simulate the old provided interface.

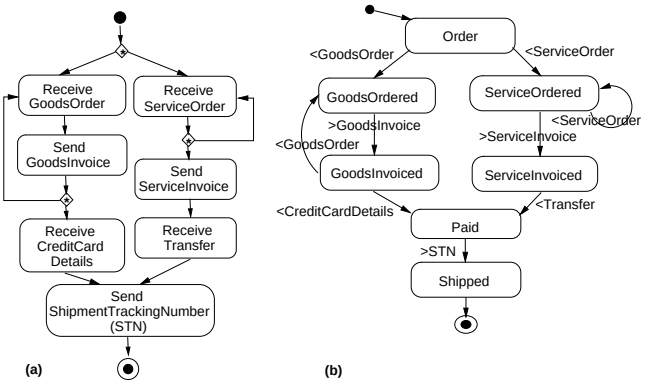

Fig. 1. Activity diagram and FSM of the provided interface.

3 Quantitative simulation

FSM modeling: In our approach we model the behaviour of a Web service interface using *Finite State Machines [6]*. Techniques exist to transform behavioral service interfaces defined in other languages (e.g. BPEL) into FSMs(see for example the WS-Engineer tool [3]). In the FSMs considered in this paper, transitions are labelled with messages to be exchanged. When a message is sent or received, the corresponding transition is fired.

An FSM is a tuple (S, L, T, s_0, F) where: S is a finite set of states, L a set of events (actions), T the transition function ($T : S \times L \longrightarrow S$). s_0 is the initial state such as $s_0 \in S$, and F the set of final states such as $F \subset S$. The transition function T associates a source state $s_1 \in S$ and an event $l_1 \in L$ to a target state $s_2 \in S$. In this model, a transition is defined as a tuple containing a source state, a label and a target state.

Figure 1 (b) illustrates the FSM of the running example which describes the behavioural interface of the service provider. We only consider the observable behaviour of a service, thus internal activities are hidden. Activities meant to send and to receive messages are modeled. The message m is denoted by $>m$ (respectively $<m$) when it is sent (respectively received). Each conversation initiated by a client starts an execution of the corresponding FSM.

We use the following notations (examples refer to the FSM depicted in the right side of the Figure 1(b)):

- $s\bullet$ is the set of outgoing transitions from s.
  (e.g. GoodsInvoiced$\bullet$ = {(GoodsInvoiced,<CreditCardDetails,Paid),
  (GoodsInvoiced, <GoodsOrder,GoodsOrdered) }).
- Label(t) is the label[1] of the transition t.
  (e.g. Label((GoodsInvoiced,<CreditCardDetails,Paid))=<CreditCardDetails)
- The *Label* operator is generalised to a set of transitions. For example, if $T = \bigcup_{i=1}^{n}\{t_i\}$ then $Label(T) = \bigcup_{i=1}^{n}\{Label(t_i)\}$; where $n =\parallel T \parallel$.
- $\parallel X \parallel$ is the cardinality of the set X.

In our previous work, we implemented an algorithm which is meant to detect the exact location of changes while comparing two FSMs P and P' (which respectively models the old provider and the new provider interfaces). A difference is detected if and only if the new interface does not simulate the behaviour of the previous interface. The outcome is a set Res of tuples (si, ti, sj, tj) where si and sj are states of P and P' respectively, while ti and tj are either null values or outgoing transitions of si and sj respectively.

Figure 2 shows three differences between P and P'. The first difference is a deletion of the operation $<$ *ServiceOrder*, which means that the new provider does not allow its client to update its service order. This difference causes an incompatibility with the required interface of the client as he can not use this operation any more. The second difference is an addition of the operation $<$ *Transfer*.

[1] In deterministic FSMs, $\forall t_1 \in s\bullet, t_2 \in s\bullet : Label(t_1) \neq Label(t_2)$.

However, this difference does not cause any incompatibility as the added operation provides a new option to its client. The third difference is the modification of the operation $>$ *STN* by the operation $>$ *ASN* (*Advanced Shipment Notice*). An incompatibility will arise because the client can not recognize this new operation.

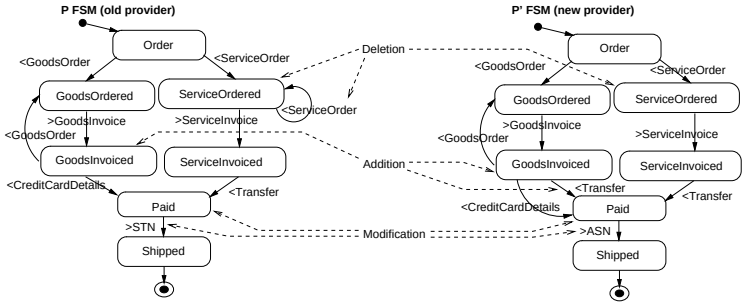

Fig. 2. Differences between the old and the new provider FSMs.

Quantitative simulation: In the detection algorithm, P and P' are traversed in parallel. A set of reached pair of states *Rps* is built such as $Rps \subseteq S \times S'$, where S is a set of P's states and S' is a set of P''s states. For each pair of states $(si, sj) \in Rps$, we compute a quantitative simulation function *Qs*. This function returns a score of differences between si's outgoing transitions and sj's outgoing transitions. $Qs : S \times S' \rightarrow [0..1]$ is defined as follows:

$$Qs((si, sj)) = \begin{cases} 1 & if, si\bullet = \{\} \\ \frac{\sum_{i=1}^{\| Diff((si,sj)) \|} Weight(D_i) + \| Label(si\bullet) \cap Label(sj\bullet) \|}{\| Diff((si,sj)) \| + \| Label(si\bullet) \cap Label(sj\bullet) \|} & otherwise \end{cases} \quad (1)$$

Where: $Diff((si, sj))$ is a set of differences pinpointed at the state pair (si, sj) such as $Diff((si, sj)) \subseteq Res$, and $D_i \in Diff((si, sj))$ for $i = 1.. \parallel Diff((si, sj)) \parallel$. The function *Weight* returns a penalty value[2] for each type of difference, and $0 \leqslant Weight(D_i) < 1$. The sum of all penalties in the state pair is added to the score of the common labels of the outgoing transitions. Common labels of the outgoing transitions of si and sj refer to the case where no difference is detected. Thus, a highest score is attributed (see (1): $\parallel Label(si\bullet) \cap Label(sj\bullet) \parallel$). To compute the quantitative simulation of the pair state, the sum of difference score and similarity score is divided by the number of these differences and similarities between the outgoing transitions of si and sj (see (1): $\parallel Diff((si, sj)) \parallel + \parallel Label(si\bullet) \cap Label(sj\bullet) \parallel$). For example, in Figure 2, if the value of the deletion penalty is set to 0.5, then the quantitative simulation is: $Qs((ServiceOrdered, ServiceOrdered)) = \frac{0.5+1}{1+1} = 0.75$.

[2] How penalty values are set is out of the scope of this paper.

Mean quantitative simulation: Once the quantitative simulation is computed to all state pairs, a mean quantitative simulation value of P and P' can be defined as follows :

$$Mqs(P, P') = \frac{\sum_{i=1}^{\| Qs \|} Qs(PS_i)}{\| Rps \|} \quad (2)$$

Where: PS_i is a pair of states such as $PS_i \in Rps$ for $i = 1..$ $\| Rps \|$. In the running example, if all the penalty values are set to 0.5 then the mean quantitative simulation is: $Mqs(P, P') = 0.875$.

4 Tests and results in BESERIAL

For validation purposes, we built a test collection consisting of 20 process scenarios from the xCBL[3] textual description of order management choreographies. These two-party choreographies describe possible document exchanges between trading partners in an Order Management business process.

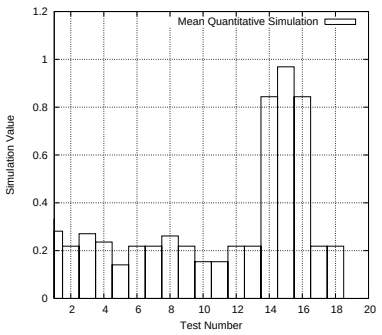

Fig. 3. Graph results of the process order collection test.

In BESERIAL[4], one interface is compared to a collection of interfaces. In Figure 3, the graph shows which interface yields less incompatibilities with respect to the interface given as reference. In this example, the interface which simulates as much as possible the given one yields a mean quantitative simulation value of 0.97. The worst result is 0.14. The interfaces in tests number 14, 15 and 16 are selected as candidates to substitute the old service.

5 Related work

Compatibility test of interfaces has been widely studied in the context of Web service conversations. Most of approaches, which focus on the behavioural dimension of interfaces, rely on similarity calculus to check, *at design time*, whether

[3] XML Common Business Library (`http://www.xcbl.org/`).
[4] http://www-clips.imag.fr/mrim/User/ali.ait-bachir/WebServices/WebServices.html

or not interfaces described for instance by automata are compatible [2]. The behavioural interface describes the structured activities of a business process. Checking interface compatibility is thus based on bi-similarity algorithms [5]. These approaches do not deal with the quantification of interface simulation.

In [6], authors introduced a technique to diagnosis message structure missmatches between service interfaces and to fix them with adapters. An extention of this technique is applied to reslove missmatches between service protocols. The proposed iterative algorithm builds a missmatch tree to help developers to choose the suitable adapter each time and incompatibility is detected. However, this technique can only be applied to protocols which describe a sequence of operations. More complex flow controls, such as loops and options, are not taken into consideration. Recent research has addressed interface similarity measures issues. In [4], authors present a similarity measure for labeled directed graphs inspired by the simulation and bi-simulation relations on labeled transition systems. The presented algorithm returns a value of a simulation measure but does not tell us more about the location of incompatibilities.

6 Future work

In this paper we focused on the calculus of the differences between two behavioural interfaces. Ongoing work aims at extending this work towards two directions: i) detecting complex incompatibilities including structural aspects, ii) guiding analysts in fixing detected incompatibilities. As we compare two different versions of a same service, we identify adequately the delta introduced by the new version. Nevertheless, if we compare two completely different services, the semantics of operations or data types must be considered.

References

1. A. Ait-Bachir, M. Dumas, and M.-C. Fauvet. BESERIAL: Behavioural service analyser. In *Proc. of the BPM Int. Conf.*, pages 374–377. Springer, 2008.
2. L. Bordeaux, G. Salan, D. Berardi, and M. Mecella. When are two web services compatible? In *Proc. of the TES Int. Conf.*, pages 15–28. Springer, 2004.
3. H. Foster, S. Uchitel, J. Magee, and J. Kramer. WS-Engineer: A tool for model-based verification of web service compositions and choreography. In *Proc. of the IEEE Int. Conf. on Software Engineering (ICSE)*, pages 771–774, 2006.
4. N. Lohmann. Correcting deadlocking service choreographies using a simulation-based graph edit distance. In *Proc. of the BPM Int. Conf.*, number 5240 in LNCS, pages 132–147. Springer, 2008.
5. A. Martens, S. Moser, A. Gerhardt, and K. Funk. Analyzing compatibility of bpel processes. In *Proc. of the Advanced Int. Conf. on Telecom. and Int. Conf. on Internet and Web Applications and Services*, pages 147–156. IEEE, 2006.
6. H. Motahari-Nezhad, B. Benatallah, A. Martens, F. Curbera, and F. Casati. Semi-automated adaptation of service interactions. In *Proceedings of the 16th International Conference on World Wide Web*, pages 993–1002. ACM, 2007.
7. J. Pathak, S. Basu, and V. Honavar. Modeling web service composition using symbolic transition systems. In *Proc. of the 21st Conf. on Artificial Intelligence Workshop on AI-driven Technologies for Service-Oriented Computing*, pages 65–80. AAAI Press, 2006.

Realizing the Internet of Things in Service-Centric Environments

Yanbo Wu

Supervised by Dr Michael Sheng

School of Computer Science,
The University of Adelaide, SA 5005, Australia
yanbo.wu@adelaide.edu.au

Abstract. "Internet of things" is a seminal vision of future technological ubiquity. It endows everyday objects with the ability to identify themselves, communicate with other objects, and possibly compute. While Radio-Frequency Identification (RFID) technologies have laid the foundation, the development of Service-Oriented Computing poses new opportunities for fully realizing the vision. The research proposal introduced in this paper proposes a methodology to realize the Internet of things in the service-centric environments. Major research challenges targeted by this methodology and possible solutions have been discussed.

Key words: Internet of Things, RFID, Service-Oriented Computing, scalability

1 Introduction

Although the phrase "Internet of Things" was first mentioned in an article of Forbes in 2002 [1], the idea was proposed by the Auto-ID Center at MIT in 2000 [2]. It was depicted as a world in which objects or people are equipped with a sensor which can report host's location, identity or some other information via a wireless network. The network connects objects and locations in the real world to information on the web and considers active participation and creation of information/content and services by citizens. This network is promised to be able to maximize the availability of objects with minimum visibility. The realization of this vision will yield a wide of range of promising benefits in diverse areas including supply-chain management, inventory control, product tracking and tracing, and human computer interaction.

To connect massive objects to large databases and networks, a simple and cost-effective system for identification is crucial. RFID (Radio-Frequency Identification) provides this functionality. Using radio frequency, the identification process is automatic since RFID does not require line-of-light to capture the information. RFID has laid the foundation for the realization of "Internet of Things". However, in order to be fully supporting it, there are still some obstacles. For example, the price is still not low enough to tag goods at item

level. To achieve lower price, the functionality of RFID tags has to be simplified. Researchers are still working on these issues. With the development of semiconductor technologies, the price is expected to be as cheap as 5 cents.

The most crucial challenge in building such a network lies in the lack of a common software fabric underlying, i.e., lack of how the softwares in the different environments can be combined to build larger, composite system. More specifically, the problem is to find a way to build a coherent application out of a large collection of unrelated software modules [5]. In recent years, Service-Oriented Computing (SOC) is emerging as a new computing paradigm for developing distributed and federated applications. Web service is a software system designed to support interoperable machine-to-machine interaction over a network [4, 3]. It is based on the Internet protocols, and on top of that, defines new protocols to describe (e.g., using WSDL) and address (e.g., using UDDI) the service instance. SOC loosely organizes the Web services and makes it a virtual network. This architecture does not require the modules of the system to be isomorphic. Therefore it's suitable to build the infrastructure for the Internet of things. In this paper, we discuss a proposal that applies SOC in realizing scalable and Internet-based RFID traceability networks. This proposal is one part of a large research effort[1].

The rest of this paper is organized as follows. Section 2 discusses the challenges in realizing the "Internet of Things". Section 3 focuses on our preliminary proposed methodology. In particular, we propose an initial architecture design and discuss solutions for several technical challenges. Finally, Section 4 concludes this paper.

2 Problem Statement

The "Internet of Things" is a technological revolution. It represents the future of communication and computing. The realization of the "Internet of Things" depends on the development of technical innovations in some important fields. One most important field is about software architecture design.

With "Internet of Things", we are able to connect everything we care in the world to the same network, to process and manage the massive collected data in this network. As a fact, Wal-Mart is generating terabytes of data everyday if tagging the goods at item level. To extend this small society to the whole world, the number of data entries will be huge. To manage such a large-scale application platform, an efficient system architecture becomes paramount important. In addition, the RFID data has the fundamental characteristics of inaccurate, dynamic, temporal and implicit inferences. To successfully realize the "Internet of Things", the following factors should be taken into consideration [5]:

- *Scalability.* This refers to a system's ability to grow in one or more dimensions such as the volume of RFID data and the number of transactions without affecting performance. Organizations that adopt RFID technology must handle data from thousands of readers distributed across various sites.

[1] PeerTrack research project, http://www.cs.adelaide.edu.au/∼peertrack/.

- *Heterogeneity.* The system may be deployed across multiple sites, companies, or even countries using different hardwares, data structures, and standards. It must support the distribution of message preprocessing functionality for example, filtering and aggregation as well as business logic across multiple nodes to better map to existing company and cross-company structures.
- *Manageability.* Good support of administration and testing is a prerequisite for the successful deployment of a solution in large-scale, distributed applications. RFID systems must facilitate the supervision, testing, and control of their individual components as well as end-to-end processing of RFID data.
- *Openness.* System interoperability is another important parameter in data integration. For instance, a well-designed reader adapter at the edge server makes the integration reader-agnostic. In addition to being hardware-agnostic, The systems should be based on existing communication protocols such as TCP/IP and HTTP as well as syntax and semantics standards such as XML, PML (Physical Markup Language[2], and EPC (Electronic Product Code[3]). An open architecture will allow use of RFID devices from a wide array of hardware providers and, more importantly, support the deployment of RFID solutions across institutional or country boundaries.

To provide the above features, the system realizing "Internet of Things" should first be a single, open architecture system for networking physical objects [2]. And it should: i) require a minimum of performance from the tag technology embedded in the objects, and ii) be flexible and adaptable to changes.

3 Proposed Approach and Methodology

Service-Oriented Computing is the computing paradigm that utilizes Web services as fundamental elements for developing applications and solutions. The services are self-described, platform-agnostic and loosely coupled. Then the service-centric environment composed by the services is naturally open, platform independent, flexible and adaptable [3, 4]. It's evident that Service-Oriented Architecture(SOA) is suitable for the application/solution level design for the "Internet of Things". In this section, we will discuss how the two ideas can be integrated.

3.1 Approach

Our approach to realize the "Internet of Things" is depicted in Figure 1. There are four types of major services in this architecture:

- *Provider Service (PS).* Each provider service provides the collected RFID data for an autonomous organization. The service can specify the data access level and format using the standard service description protocol. When a provider service joins the system, it will publish itself to both the ONS and DS.

[2] http://web.mit.edu/mecheng/pml/.
[3] http://www.epcglobalinc.org/home.

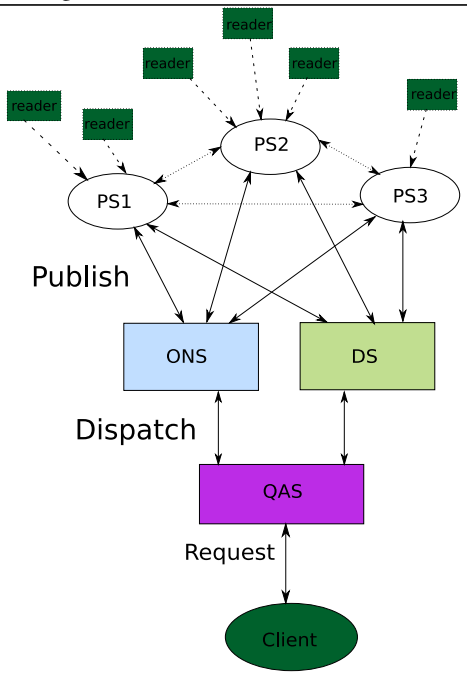

Fig. 1. A SOC architecture for Internet of Things

- *Object Naming Service (ONS).* ONS is a service to get the required object information for which only its ID is supplied from a PS where its information is stored. This is used to answer intra-service queries.
- *Discovery Service (DS).* DS is a service to break the query into sub-queries and find corresponding provider services to answer each sub-query, and then compose the answers together for the inter-service queries. Intra-service and inter-service queries will be discussed later.
- *Query Analyzing Service (QAS).* QAS is a service to analyze a query, classifying the query as either intra-service or inter-service query. QAS also directs the query to ONS or DS, based on the classification.

The whole architecture works as the following: i) the client sends a request(query) to QAS, ii) QAS analyzes the request and redirects it to either ONS or DS, iii) if it is an intra-service query, the ONS finds the corresponding PS to answer this query and binds the client and PS, iv) if it is an inter-service query, the DS breaks the query into sub-queries and tries to find all relevant provider services. Then the answer will be composed after all sub-queries are answered. During the process, provider services may need to cooperate with each other.

3.2 The Services

Services specified in the system of "Internet of Things" possess some unique characteristics. For example, provider services are software modules that offer

read-only interfaces, i.e. there are no data modification requests that can be handled. This characteristic eases the system design because there is no need to consider issues on system-level transactions.

Another important characteristic is that each service in this system represents an abstract location entity which can be a geographical information like city, state or even country, or an organization like financial department, human-resource department. Locatablity is in fact the most important feature of "Internet of Things".

3.3 Query Types

There are two main query types defined in the system, namely *intra-service* query and *inter-service* query. The former represents the queries that can be answered by the computations within one single service instance. Examples of intra-service queries are: i) "Where is object x?", ii) "When is object x last appeared at location l?", or iii) "What happened to object x when it was at location l?". In contrast, inter-service queries are used to answer questions like "Where have object x been from the time tstart to tend" or "Where did object x go when it left location x?". Those queries need two or more service instances involved in order to answer them.

Generally, it is easier to answer intra-service queries since they can be processed by individual provider services. However, it is more difficult to answer inter-service queries. To track or trace an object, there must be some interactions between individual service instances with different locations. Our solution here is to build provider services in a Peer-to-Peer (P2P) network so that the services can talk with each other to get the information. For example, when tracing an object, the Discovery Service first finds out the start point of the route and sends the request to it (the root), then the root starts a P2P query to its neighbors, when the object's current location is found, the tracing is done and the result is returned by the root. The P2P network obviously decreases the workload of the DS and leave most of the work to the provider services. Currently, we are in the process of designing such a P2P network, as well as the associated algorithms.

3.4 Data Model

The most challenging part of the design is what kind of data should be published and how should it be stored in the ONS and DS so that the query can be efficiently processed. For both types of queries, there have been some research done to ensure the performance basing on the improvement of data model about RFID systems [7, 6].

However, these data models are not adequate for the realization of "Internet of Things", especially in the service-centric environment. The data models proposed so far are largely constrained on some particular application domains such as supply chain management, which assume data shares some common properties—for example, moving together in bulk mode or having the same expiration date—and can be grouped based on such properties. In the vision of

the "Internet of Things", all different types of objects might be connected. We can not expect and assume that RFID data share the common characteristics.

In our approach, considering the proposed P2P network, we are going to use a probabilistic data model. In this model, the DS collects information only about statistics of data flow. For example, the percentage of objects moved between locations. It uses statistic information to choose neighbors for further search. Furthermore, it is possible to add more intelligence on the DS in order to accelerate the process using some existing probabilistic models such as Markov chain.

4 Conclusion

In this paper, we have proposed an initial architecture design to realize the "Internet of Things" in the service-centric environments. We discussed the technical challenges and investigated the possibility of combining Service-Oriented Computing and the RFID technology for this purpose. We advocate that a P2P, Web service-based architecture could achieve the scalability of large-scale applications such as the "Internet of Things".

The proposal reported in this paper is one part of a large research project aiming at developing a scalable, Internet-based RFID traceability network. Currently, we are investigating several technical challenges such as data models and service design on query processing. We will continue to implement the system and perform experimental studies to validate our approach.

References

1. Forbes.com, `http://www.forbes.com/global/2002/0318/092.html`
2. S. Sarma, D. L. Brock, Kevin Ashton. The Networked Physical World, Proposals for Engineering the Next Generation of Computing, Commerce and Automatic-Identification (2000)
3. M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-Oriented Computing: State of the Art and Research Challenges. In IEEE Computer, Vol 40, No 11. pp 38-45, (2007)
4. Q. Yu, A. Bouguettaya, and B. Medjahed: Deploying and Managing Web Services: Issues, Solutions, and Directions. In The VLDB Journal, Vol 17, No 3. pp 537-572 (2008)
5. Q. Z. Sheng, X. Li, S. Zeadally. Enabling Next-Generation RFID Applications: Solutions and Challenges. In IEEE Computer, September 2008, Vol 41 No 9. pp 21–28 (2008)
6. F. Wang, P. Liu. Temporal management of RFID data. In The 31st international conference on Very Large Database, Trondheim, Norway (2005)
7. H. Gonzalez, J. Han, X. Li, D Klabjan. Warehousing and Analyzing Massive RFID Data Sets. In International Conference on Data Engineering (ICDE'06), 2006, Atlanta, Georgia, USA (2006)

External Behavior Modeling Enrichment of Web Services by Transactional Constraints

Ali Khebizi
Computer Science Doctoral School
8 mai 1945 University - BP 401 Guelma 24000 - Algeria
kheali@hotmail.com

Supervised by: Hassina Seridi
Badji Mokhtar University, BP 12 Annaba 23000, Algeria
LabGed laboratory. seridi@lri-annaba.net

Abstract.

The current description of services basing on their interfaces and their protocols remains limited and does not include all the engaged interactions properties in a Web service environment. For a real deployment and a broad adoption of Web services technology, service protocols must be enriched by other properties related to interactions nature between services.
In this paper, we present a model for representing transactional effects and service protocols description improved by injection of the proposed effect model. The protocol compatibility and the equivalence analysis will be reviewed in the light of the proposed enhancements.

Keywords: Web service protocols, Transactional effects, Compensation, Compatibility analysis, Services equivalence.

1 Introduction

The current description of Web services basing on their interfaces and protocols remains limited and does not represent all the semantics of the interactions involved during services invocations. Indeed, current service protocol modeling is taking into account various characteristics which describe its external behavior (such as: order and time constraints) [1][2]. In addition, the current infrastructure of Web services (SOAP, WSDL, and UDDI) is enriched by specifications needed to manage transactions and coordination at Middleware level (such as the Frameworks: WS-Coordination [3] and WS-Transaction [4]). However consideration and management of transactional constraints in services protocols are not given the interest they deserve.
For an actual deployment and broad adoption of this technology, service protocols modeling require other enhancements. In this paper, we propose an enrichment of the external behavior of Web services description by taking into account the transactional constraints. We will also strive to study the conceptual consequences of this enrichment on compatibility and equivalence analysis.
The paper is structured as follows: In section 2, we will explain the motivations of this work. A state of the art of transactions management in service protocols is

presented in section 3. In section 4, we will propose a model for representing transactional effects and we present the involved protocol model. Section 5 will be devoted to analyzing compatibility and equivalence of protocols enriched with transactional effects. Finally, we conclude and present our future works in section 6.

2 Motivations

Because of their excessive cost and their relatively long time, transactions in Web services differ in their effects and their cancellation process from those related to traditional databases. Therefore, service providers are rather compensating transactions and affect often a part of the costs associated to customers [1]. The compensation is provided by the middleware transparently by execution of the compensation protocol predefined by the service developer [5]. In this context, it is appropriate to address a profound reflection on the following issues: How to describe and model transactions effects and their compensation effects? How to model service protocols taking into account the transactional effects? What impacts will bring the injection of transactional constraints in service protocols to the analysis of service compatibility and equivalence?
Only a few studies exist on this issue that requires more in-depth research efforts. In addition to answering previous questions, the following reasons motivate this work.
a. **Services compatibility verification:** the compatibility definition of two protocols (fully or partially), as described in [1], restricts the test criterion to operations order and to messages polarity. It has been extended to take into account the time constraints (time, date) [2]. The compatibility of two services protocols must take into account transactional aspects related to messages and their effects.
b. **To infer transactional properties for existing scenarios:** given BPEL programs availability, it is appropriate to be able to extract their transactional properties for their analysis and their manipulation. In this perspective, a BPEL program is analyzed to extract its transactional properties. Indeed, elements of management errors and transactions, such as: <*compensate*>, <*compensate scopes*> and <*compensation Handler*> constitute activities blocks related to transactional properties that must be recovered and modeled for their possible treatment.
c. Protocol consistency checking (Design Tools): In a compensation situation, it is imperative to check whether the compensation protocol is consistent with the trigger one or not? i.e.: the compensation guarantees – indeed- a "semantic cancellation" of observed effects? Transactions effects modeling and related services protocols management will provide a sound conceptual framework to check the consistency of a protocol for compensation with the trigger one.

3 Transactions management in services protocols state of the art

The state of the art of transactions management -at the protocol level- highlights four approaches that deal in a more or less rigorous way this aspect. In **Protocols languages modeling**, WSFL and XLANG languages provide extensions to the

WSDL standard, offering composition and coordination structures of services based on rules. However, no model is provided for distributed transactions management and transactions compensation is discussed in relation to data flows manipulation. This would require a considerable programming effort. In **Web transactions protocols**, the current Web services specifications are relaxing the ACID properties and strengthening mechanisms for compensation [5]. However, the majority of the proposed specifications don't deal neither with the concept of transactional effect nor with the compensations management. Both protocols have dealt with this issue are: Business Transaction Protocol (BTP) [7] and Tentative Hold Protocol (THP) [8]. In BTP, transactions effects are covered in three dimensions: *provisional effects*, *counter-effects* and *final-effects*. However, specification of effects types remains manual and specific to the engaged coordination. In addition no mechanism to ensure counter-effects consistency with effects is presented. THP is based on the reservation and allocation principle of the current transaction resources by manipulating the concepts: *attempt*, *non-blocking* and *holds reservation*. The cancellation and compensation process are then significantly reduced. But the protocol remains limited in managing the transactions effects and their manipulation. Furthermore, customers have no idea on resources they will need during the activities evolution. **Development environments of business Web services** (Enterprise Java and XML transactions) suffer from the shortcomings due to the lack of conceptual models for representing and manipulating transactional effects. Therefore, no mechanism can verify that observed effects in the real world are, really, those desired apart from traditional testing suites/scenarios. In addition, compensation is discussed in terms of a new process to execute. **The Web Service Transaction model (WSTx)** [9] proposes a WSDL language extension to describe the customer and provider's transactional behavior. However, it suffers from a deficit in modeling effects and proposes only a WSDL operations type classification following transactional criterion.
To address the deficit in effect modeling and compensation management we will propose, in what follows, a formal model for representing transactional effects which is injected thereafter in the service protocol model.

4 Modeling effects and their impact on service protocols model

The Table 1 summarizes models characteristics that bring effects representation on Web services. It presents a comparison on the basis of a set of criteria, highlighting by this their strengths and weaknesses.

Model / Criterion	OWL-S [10]	BPEL	*Colombo* [11]
Concepts	Ontology, Classes, Effects ServiceProfil, ServiceModel,	Activity, Variables, Scope Compensation	Database Query Updating, Universal Relation
Formel Meta Model	Logic description	Language	Relational Model
Concept of effect	Yes	No	Yes
Concept of State	Yes	No	Yes
Compensation handling	No	Yes	No
Formel Model for effects	**No**	**No**	**No**

Table 1: Different models representing effects Comparison

Based on the perception of effects such as query for updating the database, the *Colombo* model [11] offers advantages in effect and state concepts mastery. However, it is still failing in the management of compensating transactions and does not allow comparative effects manipulation.
We will adapt *Colombo* principle model for representing transactional effects. Indeed, in our model, effects and their compensation effects are considered as requests to update the database. Thus, a message of a service protocol will impact on the real world of a customer by executing a request to update database by type: *Insert (R), Delete (R) or Modify (R)*, where *R* is the record of the database reflecting the impact of the message on customer world. The transactional effects managing problem is reduced accordingly to that of handling query, as shown in Table 2.

Transactional effect management problem	**The corresponding Query management Problem**
Checking equivalence and difference of effects	Comparison of updating query
Finding the compensation effect for compensating an	Search a query for cancellation after
Finding elementary effects for complex effect	Queries decomposition
Cumulated effects for a complete execution path	Sequence of queries
Cumulated effects for compensation	Search a query sequence for compensation
Checking transactional effects protocols equivalence	Comparing equivalence of query sequences
Checking transactional effects protocols compatibility	Comparing sequences of query

Table 2: Transformation of effects management problem to updating queries problem

Taking into account transactional effects allows a rich representation of interactions reality in Web services. Indeed, a message will be characterized, in addition to its polarity by effects. It creates in the customer world, as well as compensation effects involved. Compensation effects are represented jointly with observed effects, in order to express the fact that service providers implement charges that differ even if effects are the same. This performance reflects the reality on the diversity of logic compensation which is specific to each provider.
New structure of message for service protocols enriched by the transactional effects: According to our model, at each message is associated a request to update the database and its corresponding complaint related to compensation effects. The new structure of a message is described as follows: ***m (p, e, e')***, where:
m: Refers to the message and its polarity *p (+,-)* as the message is input or output [1]
e: All effects observed in the customer world. This is a request to update the database.
e': All effects of compensation to defeat semantically the effects *e*. This is a request for updating the database to cancel the effects *e* while applying charges imposed by the supplier and relating to the transaction cancellation.
This modeling express in a formal way (relational queries) the effects of transactions and compensation effects for each message of service protocol.
Formal model of service protocols enriched by the transactional effects: Integrating the new structure of the message in the basic model of service protocols [1], will result in an overhaul of protocols model modeled with deterministic finite state machine. The new model protocol (*transactional effects protocol*) is described by the tuple: $\mathbf{P = (S, s_0, F, M, R, S_b)}$ where:
S: A finite set of states; s_0 Є S is the initial state of the protocol;
S_b: state of the database associated to each state of protocol;

F: The set of final states machine, with $F \subset S$; M: a finite set of messages, we associate to a message *m* two types of effects *e* and *e'*, which correspond, respectively, the requests *Ri* and *Rj* for the database updating.
$R \subset (S \times S_b)^2 \times M$: Transitions set. Each involves a state source, which is associated a database state, to a target state with its database state, following the message receipt. It should be noted: $R((s, s_b),(s', s'_b),m)$ instead of $((s,s_b), (s',s'_b),m) \in R$.
In addition, an effect function is defined, for each message allowing the transition from one state to another (with their database states), combines effects (in terms of requests) and compensations effects (corresponding requests).

5 Transactional effects protocols' compatibility and equivalence analysis

Service protocols compatibility and equivalence analysis as specified in [1] [2] should be revised in the light of proposed enhancements. Indeed, transactional effects representation will be exceeded qualitative analysis, beyond its simple syntax and structural aspect. It will be richer because it is based on semantics of transactions seen in terms of messages effects in the real world. In addition, modeling compensating effects in conjunction with observed effects representing a message by related attributes *(e, e')* expresses perfectly the real situations in which suppliers combine compensation effects for each observed effect.

✓ **Transactional effects protocols compatibility:**

Transactional effects protocols compatibility differs from that of basic protocols, due to effects induced by messages. Indeed, two compatible protocols in the basic model [1] may not be in the new context. In this sense, an interaction between two services protocols is allowed only if observed effects will be compatible. By compatibility effects, we are presuming that complaints updating request at the databases have the same type (*Delete*, *Insert* or *Modify*). This condition implies an interaction path concept redefinition to be extended to query type, as follows:

((State1.State2).Message.QueryType)*

This extension will ensure -when analyzing- verifying the compatibility of updating query and will promote a richer specification of interaction protocols between the candidates. Thus, two service protocols may be compatible only if queries -or sequences of queries- associated to messages would be compliant.

✓ **Transactional effects protocols equivalence:**

After studying various scenarios, we concluded that transactional effect protocols equivalence is conditional on final states equivalence of the two databases witch is considered on the basis of sequence equivalence of query updating.
We have identified two equivalence types for queries sequences: strict equivalence and converging equivalence.
Strict Equivalence: For each message *m* of a protocol P_1 corresponds to the corresponding message in the protocol P_2, exactly one query that it is equivalent: *i.e.* it has the same type (*Insert, Delete, and Modify*).
Bases states' converged equivalence: In this case, we are interested in the queries sequence of the complete execution paths. For each query sequence associated to a

complete execution path of a protocol P_1 corresponds to the same path in P_2, another sequence of equivalent queries. This leads to a convergence of updates inducing databases final states which are equivalent.
The two equivalence types induce two equivalence classes for transactional effects protocols: *strict equivalency Class* and *bases states' converged equivalence Class*. The second equivalence class is of particular interest because it expresses a way of achieving differently from the service providers while leading to identical databases.

6 Conclusion

In this paper, we highlighted the interest of transactional constraints modeling. These constraints have been perceived as effects affecting the customer world and have been analyzed in the context of their compensation. We proposed a model based on query for updating databases for representing transactional effects. The enriched service protocol model was presented and formalized. The second contribution is on the compatibility analysis formalization and study of transactional effects service protocols equivalence.
As future work, we plan to identify the compatibility types and to study the algorithmic aspect. We intend, moreover, the proposal for a set of operators handling transactional effects.

References

1. B. Benatallah and al : Representing, Analysing and Managing web Service Protocols. Data Knowledge Ingineering. 58 (3): 327-357, 2006
2. J. Ponge and al: Fine-Grained Compatibility and Replaceability Analysis of Timed Web Service Protocols. ER 2007: 599-614
3. F. Cabrera and al. Web Service coordination (WS-coordination), August 2005
4. F. Cabrera and al. Web Service transaction (WS-transaction), January 2004. http://dev2dev.bea.com/pub/a/2004/01/ws-transaction.html/
5. Gustavo Alonso, Fabio Casati, Hurumi Kuno,Vijay Machiraju : Web services concepts Architectures and applications, Edition Springer Verlag Berlin 2004
6. Web Services Business Process Execution Language Version 2.0 OASIS Standard, 11 April 2007, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html/
7. OASIS Business Transaction Protocol (BTP), http://oasis-open.org/committees/business-transactions/
8. J. Roberts , K. Srinivasan. Tentative hold Protocol Part 1: White Paper, W3C Note 28 November 2001, http://www.w3.org/TR/tenthold-1/
9. T. Mikalsen and al, Transactional attitudes: Reliable composition of autonomous Web Services, WDMS 2002.
10. D. Martin and al, OWL-S: Semantic Markup for Web Services, W3C Submission November 2004, http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/
11. D. Berardi and al, Automatic composition of Transition-Based Semantic Web Services with Messagging, Proceeding of 31st VLDB conference, Trondheim, Norway, 2005

A graph b-coloring based scheme for Composition-Oriented Web Services Abstraction: COWSA

Lyes DEKAR
supervised by Hamamache KHEDDOUCI

Université de Lyon, Lyon, F-69003, France; Laboratoire LIESP, Université Lyon1
Batiment Nautibus (ex.710), 43 bd du 11 Novembre 1918 F-69622 Villeurbanne Cedex, France
{ldekar,hkheddou}@bat710.univ-lyon1.fr

Abstract. We propose in this paper a self-learning scheme named COWSA, which aims to enhance the performances of existing Web services composition algorithms. Our scheme is based on a new dynamic clustering of Web services, that is oriented to Web services composition. This clustering is performed through using the b-coloring of graphs. We conduct a series of experiments to evaluate the contribution and the performances of our scheme.

1 Introduction

The Web services composition has drawn a great deal of attention recently. With the growth of the Web services number, it is essential to organize them in order to facilitate the discovery of services participating to the composition process. In this paper, we propose to regroup the Web services according to the compositions made by the users (through their requests for composed services). This approach seems to be a new approach for the Web services classification since the majority of Web services classification methods [3] is based on the similarity between Web services. To achieve our clustering, we use a b-coloring of graphs [2]. The b-coloring can be defined as follows : Let $G = (V, E)$ be an undirected connected and simple graph with a vertex set V and an edge set E. The b-coloring of G is a vertex coloring function c from V to the set of colors $\{1, 2, ..., k\}$ such that: 1- for each pair of adjacent vertices $(v_i, v_j) \in E$, $c(v_i) \neq c(v_j)$ (proper coloring). 2- In each color class, there exists at least one vertex having neighbors in all other color classes. Such a vertex is called a *dominating vertex*. A color that has a dominating vertex is called a *dominating color*.

2 The Composition-Oriented Web Services Abstraction: COWSA

In this paper, we identify two kinds of services: *atomic services* and *composite services* (CS). An atomic service is a Web service that fulfils requests without

depending on other Web services. A composite service is a Web service that includes a set of atomic services, called *component Web services*, and can be itself a part of another composite service. We define a *complex request* as a user's request that consists of a set of service functions $F_1, ..., F_k$, which cannot be satisfied by one atomic service. A complex request is satisfied by a composite service. In order to satisfy a complex request, a composite service should be constructed through the Web services discovery and composition operations. The aim of this paper is to propose a method to organize the Web services in such a way to enhance the performances of the web services composition methods. Our proposed method consists of regrouping Web services that are often composed together in clusters. A set of services is assumed often composed together if they appear often in a same composite service when satisfying users complex requests. This approach aims to identify stable services sets, which contain services that are regularly and frequently composed together. Then, we obtain an *Abstraction* of Web services, where every identified set of services represent a template of composite services, or what we name a *meta composite service* (MCS). A MCS exhibits a WSDL interface and can be invoked as an atomic service. The MCS are used by the Web services composition methods to accelerate the discovery of component Web services that are required in the composition process. Then, the MCS are used to fulfill the users complex requests. Since the users requests can change over the time, then the MCS can change too. Therefore, we propose to give our method following a self-learning model.

We model our system by an undirected edge-weighted graph $G = (V, E)$, where V is the vertex set and E is the edge set. Vertices in G represent services, edges correspond to the composition relation between services, and edge weights represent the number of times two linked services are composed. This information is presented in the Composition Weight Matrix (CWM). The clustering here consists to regroup services such that we obtain a large intracluster composition weight and a small intercluster composition weight.

The filtered graph: In order to regroup the services joined by a large weighted

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0								
S_2	2	0							
S_3	1	13	0						
S_4	12	3	2	0					
S_5	5	14	23	3	0				
S_6	9	14	4	5	15	0			
S_7	1	17	6	0	9	12	0		
S_8	12	19	1	18	12	4	14	0	
S_9	13	6	17	20	3	2	16	2	0

Fig. 1. A composition weight matrix (CWM).

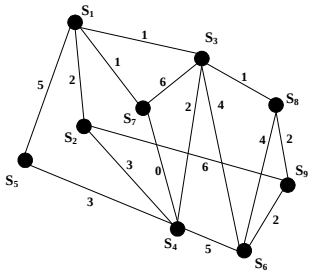

Fig. 2. The filtered graph.

link in the same cluster, we remove all edges with a weight larger than a threshold α. Consequently, after removing these edges, we obtain a *filtered graph* $G_{<\alpha} = (V, E_{<\alpha})$, such that $E_{<\alpha} = \{(v_i, v_j) \mid CWM(v_i, v_j) < \alpha\}$. Figure 2 gives the filtered graph $G_{<9}$ corresponding to the composition weight matrix given in Figure 1.

3 A partial dynamic algorithm for a b-coloring of graphs

The proposed algorithm is composed of two parts. The first part of our algorithm constructs a b-coloring to obtain a partition of the filtered graph $G_{<\alpha}$ into disjoined color classes $\{C_1, C_2, ..., C_k\}$ that represent clusters. The second part of our algorithm maintains this clustering when edges are added or removed from the filtered graph. Then, the proposed algorithm is a partial dynamic algorithm. Before presenting the algorithm, we first give some notations and definitions. We let Δ be the maximum degree of G and $c(v)$ be the color of the vertex v in the graph G. For every vertex v, we define $N(v)$ its open neighborhood, as the set of vertices adjacent to v. The set of colors of $N(v)$ is denoted $N_c(v)$. We note L the color set used in the graph. For each color c used in the graph, we associate a variable $Dom[c]$ that indicates if the color c is a dominating or a non-dominating color (*true*: if c is dominating. *false*: otherwise). Finally, we define a function *weight(v,c)* that indicates the *composition weight* between the vertex v and the color c. This function is defined by: $weight(v, c) = \max\{CWM(v, v') \mid c(v') = c\}$.

3.1 The clustering construction algorithm

In this subsection, we give the first part of our algorithm, which performs the b-coloring and constructs the clusters. The b-coloring is made in two steps:
***Procedure 1: the coloring initialization*:** In the first procedure, the graph is initialized by coloring it with a maximum number of colors ($\Delta + 1$). The procedure starts by coloring the vertex having maximum degree Δ by the color 1 and adds it to a list S. Then, the procedure color the remaining vertices as follows: the vertex v_i with the largest degree among all colored vertices belonging to S is selected. If there is non-colored vertices v_j adjacent to v_i then a new color is assigned to every one of them and are added to S. The assigned color must be different from those appearing in the neighborhood of v_j or v_i, and must not Exceed $\Delta + 1$. Finally, the procedure checks if the color of v_i is a dominating color. In this case, this color is marked as dominating. After that, the vertex v_i is removed from S. The operation is repeated for every colored vertex until all the graph is colored. Let us consider the filtered graph $G_{<9}$ obtained in Figure 2. By performing the procedure 1, the filtered graph $G_{<9}$ has an initial coloring with a maximum number of colors $\Delta + 1 = 6$, as shown in Figure 3. Among this colors, only the vertex S_4 is a dominating vertex, and then only the color 1 is dominating ($Dom[1] = true$).

Procedure 2: find a b-coloring of G**:** In the coloring obtained after the execution of the previous procedure, some colors could be not dominating. Then, the Procedure 2 finds a b-coloring of a graph G where all the colors are dominating. Hence, the strategy consists to remove a non-dominating color p from the graph by recoloring every vertex colored with p by an already used color not appearing in its neighborhood. If there is choice between many colors, then the color that has the largest composition weight with the vertex is selected. After that, the procedure checks if there is non-dominating colors that have dominating vertex. The operation is repeated until all the colors are marked as dominating. Let us consider the colored graph obtained after the execution of the procedure 1 and given in Figure 3. By performing the procedure 2, we obtain a b-coloring of a graph $G_{<9}$, as shown in Figure 4. We can observe that four colors appear in the

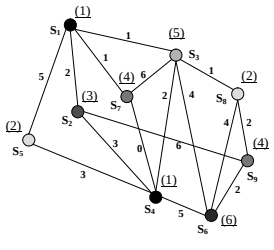

Fig. 3. A graph $G_{<9}$ Coloring initialization.

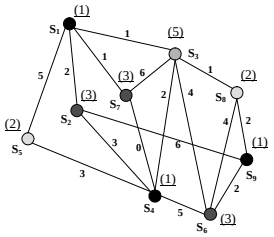

Fig. 4. the b-coloring of a graph $G_{<9}$.

graph $G_{<9}$. This means that the Weighted composition graph is partitioned into four color classes. The composition weight between the services that are in the same cluster (color class) is larger than α.

3.2 The clustering maintenance algorithm

The composition weights between services can evolve in the time because the self-learning of our method. Then, an edge can appear (resp. disappear) in (resp. from) the filtered graph if its weight becomes under the threshold value (resp. its weight exceeds the threshold value). Then, we propose an edge-dynamic algorithm to maintain the b-coloring when edges are added or removed from the graph. We assume in this algorithm that if a color class contains several dominating vertices, then only one of them represents this class. This one is called *representative dominating* vertex. A representative dominating vertex x is said *satisfied* if for any color q in the graph, there exists at least one vertex $y \in N(x)$ such that $c(y) = q$. If there exists only one such a vertex then this one is called a *Satisfaction vertex*. Any change of the satisfaction vertex color can affect the b-coloring. The vertices that are neither representative dominating vertices nor satisfactions vertices are called *Normal vertices*.

The edge adding When an edge (v, y) is added to the graph, we can distinguish three different cases, according to the endpoints of the added edge:
1- The edge is added between two vertices having different colors: In this case, the b-coloring of the graph is not affected. **2- The edge is added between a normal vertex v and another vertex having the same color**: In this case, the coloring is not anymore proper, and the b-coloring conditions are not verified. Then, the color of the normal vertex must be changed. Hence, we can distinguish four different cases: (a) *The vertex v is adjacent to a dominating vertex of every color in the graph*: In this case, we give the vertex v a new color for which v will be dominating. (b) *There exists at least a color c to which the vertex v is not adjacent*: In this case, we give the vertex v this color. (c) *The vertex v is adjacent to all the colors in the graph, and there exists at least one color c that appears only on normal vertices w*: in this case, the vertex v takes the color c, which causes a not proper coloring. Then, the vertex w takes another color c' not appearing in its neighborhood. (d) *The vertex v is adjacent to satisfaction and/or dominating vertices with every color in the graph and there exists at least one color c that does not appear on a dominating vertex adjacent to v, but on satisfaction vertices w adjacent to v*: in this case, the vertex v takes the color c. Hence, the coloring is not anymore proper. Then, we give to the satisfaction vertex w another color not appearing in its neighborhood. The color change of w implies that the dominating vertex x that was satisfied by w is not anymore satisfied. If this dominating vertex is the only one for its colors then the b-coloring is not satisfied. In order to reestablish the b-coloring without systematically removing the non-dominating color, we try to put the previous color of w on another normal vertex z adjacent to v (by respecting a proper coloring) to reestablish the dominating condition. If such a vertex does not exist then we remove the color $c(x)$ from the graph and we color every uncolored vertex with the smallest color not appearing in its neighborhood. **3- The edge is added between a satisfaction vertex v and another satisfaction or representative dominating vertex y having the same color**: The coloring is not proper and then the color of one of the endpoints vertices must be changed. Then, we change the color of a satisfaction vertex v. Hence, the dominating vertex x satisfied by v does not respect anymore the dominating condition. Therefore, we make the same actions as in the point (d) of the previous case.
The edge removing If an edge between a unique dominating vertex v of a color c in the graph and one of its satisfaction vertices y is removed, then v is not anymore dominating. Hence, we perform the same actions as in the point *(d)* of the second case of edge adding.

4 Experiments

In this section, we evaluate the performance and the contribution of our scheme. We implement the Web services composition method given in [1]. Then, we compare the performances of this method with the same method using our scheme. Two metrics are defined to evaluate the performance of our scheme: the *Clus-*

ters stability rate and the *average search time* (AST). The clusters stability rate is the ratio of Web services that change cluster during an interval of time Δt. The average search time AST is the average time required by the service search engine of the composition process to find all the Web services implied in the composition. Figure 5 shows the clusters stability rate according to the simulation

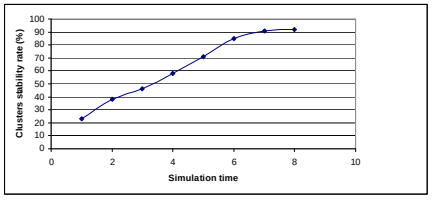

Fig. 5. Clusters (Meta composite services) stability rate vs. Simulation time.

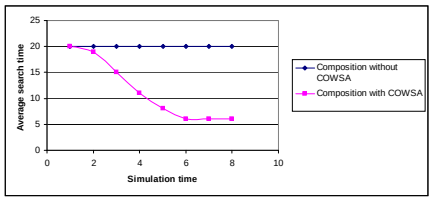

Fig. 6. The average search time vs. Simulation time.

time. We can observe that the cluster stability rate increase with time. This is explained by the self-learning process of COWSA, which enables it to learn more about the behavior of users in the construction of their complex requests. Then, the clusters become more stable and less sensitive to new composite services. Figure 6 shows the average search time (AST) of the Web services composition method given in [1] with using COWSA and without using it. We can observe that from the third time, the AST of the composition using COWSA becomes less than the AST of the composition not using COWSA. We can explain this behavior by the self-learning process of COWSA appearing in Figure 5.

5 Conclusion and future works

In this paper, we present a composition-oriented Web services abstraction scheme, called COWSA. COWSA is a self-learning method that aims to enhance the performances of Web services composition methods.

References

1. Pat. P. W. Chan and M. R. Lyu. Dynamic web service composition: A new approach in building reliable web service. *22nd International Conference on Advanced Information Networking and Applications*, pages 20–25, 2008.
2. B. Effantin and H. Kheddouci. A distributed algorithm for a b-coloring of a graph. *International Symposium on Parallel and Distributed Processing and Applications (ISPA-2006), Lecture Notes in Computer Science*, 4330:430–438, 2006.
3. S. RAM, Y. Hwang, and H. Zhao. A clustering based approach for facilitating semantic web service discovery. *15th Annual Workshop on Information Technolgies and Systems (WITS)*, 2006.