SERVICE COMPUTATION 2011

The Third International Conferences on Advanced Service Computing

ISBN: 978-1-61208-152-6

September 25-30, 2011
Rome, Italy

SERVICE COMPUTATION 2011 Editors

Hideyasu Sasaki, Ritsumeikan University - Kyoto, Japan

Guadalupe Ortiz Bellot, University of Cádiz, Spain

Bernhard Hollunder, Hochschule Furtwangen University - Furtwangen, Germany

SERVICE COMPUTATION 2011

Foreword

The Third International Conferences on Advanced Service Computing [SERVICE COMPUTATION 2011], held between September 25 and 30, 2011, in Rome, Italy, continued a series of events targeting service computation on different facets. It considered their ubiquity and pervasiveness, WEB services, and particular categories of day-to-day services, such as public, utility, entertainment and business.

The ubiquity and pervasiveness of services, as well as their capability to be context-aware with (self-) adaptive capacities posse challenging tasks for services orchestration, integration, and integration. Some services might require energy optimization, some might requires special QoS guarantee in a Web-environment, while other a certain level of trust. The advent of Web Services raised the issues of self-announcement, dynamic service composition, and third party recommenders. Society and business services rely more and more on a combination of ubiquitous and pervasive services under certain constraints and with particular environmental limitations that require dynamic computation of feasibility, deployment and exploitation.

We take here the opportunity to warmly thank all the members of the SERVICE COMPUTATION 2011 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SERVICE COMPUTATION 2011. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SERVICE COMPUTATION 2011 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SERVICE COMPUTATION 2011 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of service computation.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the charm of Rome, Italy.

SERVICE COMPUTATION 2011 Chairs:

Ali Beklen, IBM Turkey, Turkey
Emmanuel Bertin, Orange-ftgroup, France
Steffen Fries, Siemens Corporate Technology - Munich, Germany
Bernhard Hollunder, Hochschule Furtwangen University – Furtwangen, Germany
Paul Humphreys, Ulster Business School/University of Ulster, UK
Arne Koschel, Fachhochschule Hannover, Germany

SERVICE COMPUTATION 2011

Committee

SERVICE COMPUTATION Advisory Chairs

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Mihhail Matskin, KTH, Sweden
Hideyasu Sasaki, Ritsumeikan University - Kyoto, Japan
Bernhard Hollunder, Hochschule Furtwangen University – Furtwangen, Germany
Paul Humphreys, Ulster Business School/University of Ulster, UK
Arne Koschel, Fachhochschule Hannover, Germany
Michele Ruta, Politecnico di Bari, Italy

SERVICE COMPUTATION 2011 Industry/Research Chairs

Ali Beklen, IBM Turkey, Turkey
Mark Yampolskiy, LRZ, Germany
Steffen Fries, Siemens Corporate Technology - Munich, Germany
Emmanuel Bertin, Orange-ftgroup, France

SERVICE COMPUTATION 2011 Technical Program Committee

Silvana Aciar, Universidad Nacional de San Juan, Mexico
Tanovic Anel, BH Telecom d.o.o. Sarajevo, Bosnia and Herzegovina
Irina Astrova, Tallinn University of Technology, Estonia
Ali Beklen, IBM Turkey, Turkey
Emmanuel Bertin, Orange-ftgroup, France
Ismailcem Budak Arpinar, University of Georgia, USA
Radu Calinescu, Aston University, UK
Florian Daniel, University of Trento, Italy
Leandro Dias da Silva, Federal University of Alagoas - Maceió, Brazil
Erdogan Dogdu, TOBB University of Economics and Technology - Ankara, Turkey
Schahram Dustdar, Vienna University of Technology, Austria
Steffen Fries, Siemens Corporate Technology - Munich,, Germany
Jaafar Gaber , Université de Technologie de Belfort-Montbéliard (UTBM), France
Luis Gomes, Universidade Nova de Lisboa / UNINOVA-CTS - Monte de Caparica, Portugal
Gustavo González-Sánchez, Mediapro Research – Barcelona, Spain
Victor Govindaswamy, Texas A&M University-Texarkana, USA
Mohamed Graiet, Institut Supérieur d'Informatique et de Mathématique de Monastir, Tunisie
Bernhard Hollunder, Hochschule Furtwangen University - Furtwangen, Germany
Paul Humphreys, Ulster Business School/University of Ulster, UK
Mirjana Ivanovic, University of Novi Sad, Serbia
Jinlei Jiang, Tsinghua University - Beijing China
Timothy K. Shih, Asia University - Wufeng, Taiwan
Arne Koschel, Fachhochschule Hannover, Germany
Natalia Kryvinska, University of Vienna, Austria

Copyright Information

Table of Contents

Collecting Factors for Motivating Energy-Saving Behavior

Marc Jentsch, Marco Jahn, René Reiners, Uwe Kirschenmann
Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven
D-53754 Sankt Augustin, Germany
{*marc.jentsch, marco.jahn, rene.reiners, uwe.kirschenmann*}*@fit.fraunhofer.de*

***Abstract*—Ubiquitous computing systems are deployed for support in many different domains, one of them is energy efficiency. Existing ubiquitous computing systems, which aim at helping people to save energy, usually give feedback about their users' energy consumption. But these systems do not take into account age or gender of the users coming with different attitudes and motivation. Since we are considering user awareness as important factor for the success of pervasive energy saving support systems, we are going to develop a system on our own. In this paper, we investigate what requirements our system must meet and how it should be designed. We explore people's knowledge about energy efficiency and the motivation to save energy. Then, we investigate the impact of age on these aspects. The target groups are female, divided into two age groups ranging from 13 to 15 years and 23 to 30 years. Finally, we draw conclusions how to design our user-adaptive energy saving support system.**

***Keywords*-energy management, user interfaces, context adaptation, ubiquitous application;**

I. Introduction

Today's energy consumption behavior is characterized by a high amount of wasting. This causes unnecessary costs for companies or private households. In addition, global warming is being boosted, because more energy has to be produced which leads to higher CO_2 emissions. The reason for this behavior is often a lack of knowledge about the current energy consumption. In many cases, it is hard to identify energy-wasting devices. The only feedback about the own energy consumption is a yearly bill only allowing a retrospective, summarized view. This makes a change of behavior difficult due to the lack of a direct connection between cause and effect.

Current feedback-based ubiquitous systems are mainly based on the results from early studies in the field of environmental psychology that experiment with different interventions like feedback, goal-setting and rewards. Abrahamse et al. [1] provide a review of such studies, most of them resulting in energy savings up to 15%. But, Abrahamse et al. also note that most of these studies did not take into account determinants like knowledge, attitude, age etc. and that little is known about the long-term effects of the applied interventions. Other analyses come to similar results regarding the power of constant real-time feedback [2], [3] but also mention that environmental, social, and educational factors need to be considered when aiming at behavioral changes.

Since demographic aspects influence energy consumption behavior [4], we state that applications must be customized for different population groups. Our goal is to develop a pervasive system that supports people in saving energy. Unlike existing approaches, our system shall be context aware by taking the individual user into account. In this paper, we want to find ideas how to design our energy saving support system which adapts to different user groups. We have conducted a study that finds out about some differences in attitude and motivation towards the topic of saving energy with regard to different demographic aspects. This serves as basis for design decisions for our ubiquitous energy saving support system.

As starting point for investigation we take individual differences into consideration, as it is known from psychology. As this research topic is rather complex, the focus here is on age differences in prosocial behavior and will continue with gender differences in a following study. In this sense, Baldassare and Katz [5] show that women are significantly more likely to engage in environmental practices. That is the reason why we start investigating female behavior. Aging has a direct impact on energy consumption as the attitude tends to change over lifespan [6]. Therefore, we investigate possible differences between younger and older women with regard to energy saving and discuss how our energy saving support system should be individually designed.

The remainder of the paper is structured as follows. In Section II, we discuss related work which is occupying with the psychological background about motivational aspects to save energy. Additionally, we present existing feedback-based systems. In Section III, the setup of the study is described, its results follow in Section IV, the discussion in Section V. An outlook on future work is given in Section VI.

II. Related Work

Blocket et al. [7] analyzed the 1993 General Social Survey and found out that women show more environmental concern than men. On the other hand, they are not more likely to engage in environmental action. Steel [8] performed four waves of mail surveys and telephone follow-ups to

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-152-6

investigate the link between environmental attitude and behavior. As a result, women are more likely to participate in environmentally protective behavior. This is confirmed by Stern et al. [9] who found that women express stronger intentions to act than men. And later they found females having stronger biospheric-altruistic values [10]. Schahn et al. [11] again confirmed women having more environmental concerns. At the same time, men showed higher knowledge about environmental problems in their study.

A survey [12] of 801 women at 18 years or older addressing attitudes and awareness about energy emphasizes the importance of the role of women when it comes to energy awareness. For example, 77% of the women take primary or equal responsibility for paying their electricity bills. 97% try to save energy, e.g., by turning off lights, using energy saving lamps etc.

Almost all industrial and scientific efforts to increase energy awareness of users are based on the concept of providing constant real-time energy consumption information. Several products focussing on feedback are currently entering the market, e.g. Google PowerMeter [13], Microsoft Hohm [14], Greenbox [15] or The Power Tab [16]. As an example from the scientific community, Mattern et al. [17] present the eMeter system, which connects a smart meter with a mobile phone to provide the user with real-time feedback on device level. The user interface visualizes consumption data as well as historical data and costs per device. The power-aware cord [18] is a classical example for a ubiquitous computing system that aims at enhancing energy awareness: It is an electrical power strip that visualizes the amount of energy passing through it by animating the wire with lights. Jacucci et al. [19] present the EnergyLife system, which combines real-time feedback information with goal setting and awareness tips.

All systems miss to take into account the individual social and educational background. They show consumption data, costs and CO_2 emission and assume that the same visualization works equally well for different gender or age, associated with different attitude and motivation of users. In this paper, we will show that different user groups come up with different attitude and motivation towards energy saving. Additionally, we will draw conclusions how our own user adaptive energy saving support system can be designed.

III. Study

In order to find out how the age of participants influences motivation and knowledge about energy consumption, we created a questionnaire that we gave to two groups at different ages. We expected to find ideas how to design our user-adaptive energy saving support system. As related work promises women to be higher motivated to save energy, we started comparing girls and women.

The survey was conducted in two sessions, each lasted 30 minutes. In the first session, we engaged eight girls in the age of 13 to 15 representing teenagers. The second session contained eight women at the age of 23 to 30 representing grown-ups. All participants had never used any energy saving support system. At the beginning, both groups were briefed that we are investigating knowledge and motivation regarding energy saving. We explained that the questionnaire was anonymous and they will not be judged in order to decrease potential inhibitions. Each participant got one questionnaire. A moderator read aloud every question which the participants should answer. This procedure had the advantage of clarifying arising issues together with the participants.

The first three questions investigated the background knowledge regarding energy savings in order to find out what kind of information needs to be provided by our assisting system. First, we wanted to know if there is awareness of energy consumption behavior at all. Then, we asked for methods for energy saving without predefined options. Next, we gave predefined options and a free text field for finding out the source of this knowledge.

The latter two questions asked for the motivation for saving energy. From the gathered results, we expected to draw conclusions on how we can increase peoples' motivation. The first question "*What can be reasons for you or others to save energy?*" targeted to find every possible motivation in general. The whole group should name possible motivations for energy saving aloud. The second question asked each participant individually to rank the collected aspects by personal priority. Reasons that were personally not relevant should be listed unranked in a special area of the questionnaire. By this, we expected to get an overall ranking of the importance of motivational aspects.

IV. Results

In this section we present and arrange the results of the questionnaire which will then be discussed in Section V.

Awareness: All participants answered that they have reflected how energy can be saved. At the same time, everyone stated to try to save energy at least once in a while.

Knowledge: With respect to the question how energy can be saved we identified two classes of answers. On the one hand the participants proposed to change equipment. On the other hand, a change of behavior was proposed. Each class could be divided into four subclasses.

The equipment class consisted of using *Energy Saving Lamps*, buying *Efficient Devices*, *Thermal Insulation* and *Solar Plants*. For the *Efficient Devices* subclass the women suggested to pay attention to energy efficiency when buying new devices in general or gave examples like new washing machines or new refrigerators.

In the behavior class, the participants argued to completely remove unused devices from the power line instead of using the stand-by mode. They also stated to switch off lights when not being in the room. We summarized further answers as

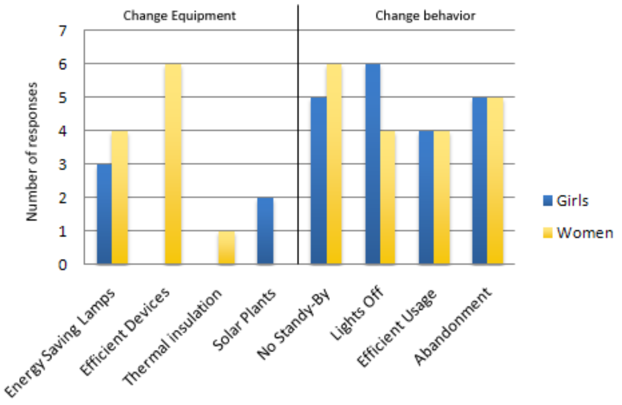


Figure 1. Grouped answers to the question: ”*How can energy be saved?*”

deliberate *Efficient Usage*. Examples are ”*Close the window when heating*” and ”*Don't start the dish washer before it's full*”. For the last subclass the participants proposed to abandon e.g. a dryer, the elevator or to live vegan. Figure 1 illustrates the distribution of answers to the question about ways to save energy.

Knowledge Sources: Figure 2 shows the distribution of the answers to the question ”*How do you know about energy saving?*” All four girls who filled in the *Misc* field wrote: ”*I have found out myself*”. One of the women who filled in the *Misc* field wrote: ”*By usage*”, the other one answered ”*Radio, Friends*”.

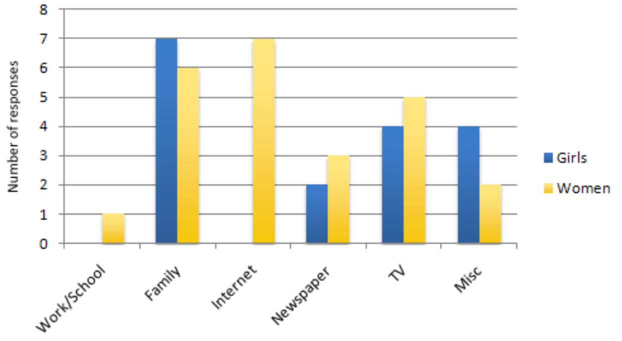


Figure 2. Answers to the question: ”*How do you know about energy saving?*”

Motivation: During the group discussion, the girls identified six motivational factors for saving energy: *Environment Protection*, saving *Costs*, parental *Bringing-Up*, *Side Effects*, *Personal Visualization* and *Games*. *Side Effects* was explained using the example of vegetarian lifestyle which has a lower energy need than non-vegetarian lifestyle. In this example, the motivation to live vegetarian was to save animals while saving energy is just a side effect. *Personal Visualization* means that if the girls just see how much energy they consume, this would be a reason to try to save energy.

Figure 3 shows how often the reasons for saving energy were put on which rank by the girls. For example, *Environment Protection* was classified by six of the girls on the first rank, placed by one girl on the third and another one on the fourth. The diameter of each circle is proportional to its value. The idea to come up with *Games* where the goal is to save energy was seen as promising during the group discussion but in the end none of the girls ranked it as relevant.

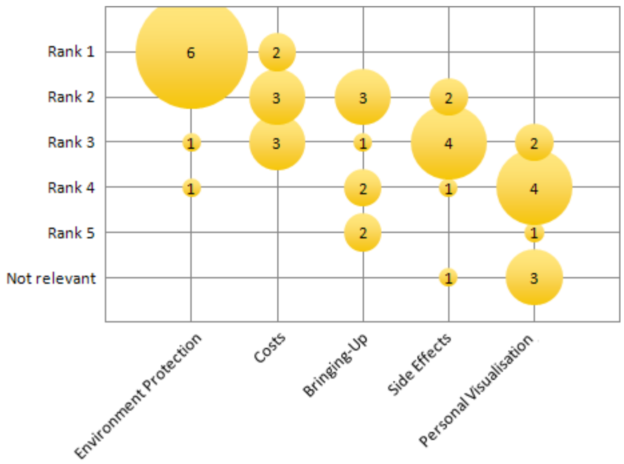


Figure 3. Overview of girls' motivation rankings

The women came up with eight motivational factors: Saving *Costs*, *Environment Protection*, *Social Pressure*, *Laws*, *Less Attrition*, *Raising Awareness*, being *Trendy* and being *Opposed to Energy Companies*. *Laws* refer to the assumption that if more laws like the prohibition of traditional light bulbs had been adopted, less energy would be wasted. *Less Attrition* means that devices are better preserved when being less used. The women also argued that starting to save energy would raise the self-awareness which is a good motivation. Figure 4 shows the women's ranking distribution. Being *Trendy* and being *Opposed to Energy Companies* was not ranked as relevant by any of the women.

V. Discussion

In this section we discuss the results of Section IV and conclude how to design our energy saving support system.

Awareness: The fact that everyone in our study states to have reflected about how energy can be saved shows that there is a general awareness of energy consumption. Also everyone answered that she tried to save energy. So, in general the motivation to save energy exists and does not need to be arisen by our system. This result may be biased by the fact that being motivated to save energy is socially requested.

Knowledge: When comparing the answers how energy can be saved, the most obvious difference between the groups is that six of the women mentioned to pay attention to energy efficiency when buying new devices while none

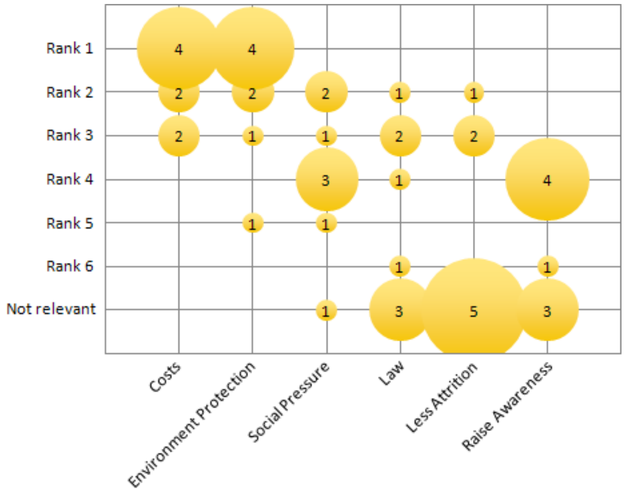


Figure 4. Overview of women's motivation rankings

of the girls had this idea. This is probably because girls do not have to buy new devices. As a conclusion, our energy saving support system will not suggest any hints to younger users because this would be a useless tip for them. It only might lower the acceptance of the system. For grown-up users, these kind of hints will be included in the system.

Two of the girls mentioned *Solar Plants*. Although, solar plants do not help to save energy, our system could show how the energy is produced.

As for the other answers the difference between the groups is two at maximum, we do not consider any distinction for different aged users of our system on these topics.

Since the answers *No Stand-By* and *Lights Off* were named very often, we interpret that the knowledge about these energy saving options is omnipresent. So, our system will not introduce them, small reminders will suffice. In contrast, information on rarely mentioned alternatives, like *Thermal Insulation*, needs to be introduced.

Knowledge Sources: With respect to the probands' knowledge about energy saving, there are again only two categories with a difference between the groups. The fact that nearly all of the women but none of the girls know about energy saving from the *Internet* may be biased by the fact that some of the women work as computer scientists. That is why we don't make conclusions for our system in this case.

The second difference is that much more girls think they worked out energy saving methods by themselves. This may indicate missing awareness of the amount of external information they are exposed to on a daily basis.

Regarding the sources *Work*, *School*, *Family* and *Friends* the information is rather pushed to the user. Coming to the sources *Internet*, *Newspaper*, *TV* and *Radio* users rather select which information to get, so these are pull services. Summarized, women named eight push and 16 pull sources while girls wrote eight push and seven pull sources. Since people have complete freedom to choose pull services, we conclude that pull services are prefered by women. In contrast, as the younger participants of our study do not use pull services regularly, information rather needs to be pushed. Hence, our energy saving support system will rather push information to younger users but provide information to be actively requested for older ones.

As interesting overall result, the family is a knowledge source for nearly everybody while work resp. school is not. While being prominent at home, saving energy does not seem to be covered at work and at school. So, we will deploy our energy saving support system in work and school environments.

Motivation: The main motivational aspects were named by both groups. Besides *Environment Protection* and *Costs*, *Bringing-Up* is similar to *Social Pressure* and *Personal Visualization* denotes the same factor as *Raise Awareness*. *Law* and *Less Attrition* were only mentioned by the women, while *Side Effects* was stated solely by the girls.

The motivational aspects' ranking differs more. While the women ranked *Environment Protection* and *Costs* almost equal, the girls gave a much higher priority to *Environment Protection*. We conclude this is again because girls do not have to pay the energy bill by themselves. Our energy saving support system will keep in mind what motivates the user group and provide the appropriate incentives.

In general, *Environment Protection* and *Costs* are the most important motivational aspects since everyone ranked one of them first. On the other hand, the approach to use *Games* to get people to save energy does not seem to be promising since nobody felt attracted. The same applies to the attempt to make energy saving trendy and ways to incite against energy companies.

VI. Future Work

With the conclusions drawn from this survey we derived features that we will implement in our adaptive energy saving support system. Next, we will set up the same study with appropriately aged groups of men to investigate gender as other influencing factor. After that, we will implement adapted systems for each group and evaluate them in user tests.

Acknowledgements

This research was supported by the European Commission within the SEEMPubS project (project No. 260139).

References

[1] W. Abrahamse, L. Steg, C. Vlek, and T. Rothengatter, "A Review of intervention studies aimed at household energy conservation," *Journal of Environmental Psychology*, vol. 25, pp. 273–291, 2005.

[2] G. Wood and M. Newborough, “Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design,” *Energy and Buildings*, vol. 35, no. 8, pp. 821–841, 2003.

[3] S. Darby, “The effectiveness of feedback on energy consumption,” *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, no. April, 2006.

[4] D. Newman and D. Day, *The American energy consumer*. Ballinger, 1975.

[5] M. Baldassare and C. Katz, “The personal threat of environmental problems as predictor of environmental practices,” *Environment and Behavior*, vol. 24, no. 5, p. 602, 1992.

[6] E. Yamasaki and N. Tominaga, “Evolution of an aging society and effect on residential energy demand,” *Energy Policy*, vol. 25, pp. 903–912, 1997.

[7] T. Blocker and D. Eckberg, “Gender and environmentalism: Results from the 1993 general social survey,” *Social Science Quarterly*, vol. 78, pp. 841–858, 1997.

[8] B. Steel, “Thinking globally and acting locally?: environmental attitudes, behaviour and activism,” *Journal of Environmental Management*, vol. 47, no. 1, pp. 27–36, 1996.

[9] P. Stern, T. Dietz, and L. Kalof, “Value orientations, gender, and environmental concern,” *Environment and behavior*, vol. 25, no. 5, p. 322, 1993.

[10] P. Stern, L. Kalof, T. Dietz, and G. Guagnano, “Values, Beliefs, and Proenvironmental Action: Attitude Formation Toward Emergent Attitude Objects1,” *Journal of Applied Social Psychology*, vol. 25, no. 18, pp. 1611–1636, 2006.

[11] J. Schahn and E. Holzer, “Studies of individual environmental concern: The role of knowledge, gender, and background variables,” *Environment and Behavior*, vol. 22, no. 6, p. 767, 1990.

[12] G. Q. R. Research, “Women’s Survey on Energy & the Environment,” 2009.

[13] Google, “Google PowerMeter,” 2011, available online at http://www.google.org/powermeter; visited on June 30th 2011.

[14] Microsoft, “Microsoft Hohm,” 2011, available online at http://www.microsoft-hohm.com; visited on June 30th 2011.

[15] SilverSpring Networks, “greenbox - Empowering energy customers to better manage their usage,” 2011, available online at http://getgreenbox.com; visited on June 30th 2011.

[16] energyaware, “The PowerTab In-Home Display,” 2011, available online at http://www.energy-aware.com/our-products/ihd; visited on June 30th 2011.

[17] F. Mattern, T. Staake, and M. Weiss, “ICT for green,” in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking - e-Energy ’10*. Passau, Germany: ACM Press, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1791314.1791316

[18] A. Gustafsson and M. Gyllensward, “The power-aware cord: energy awareness through ambient information display,” in *Computer Human Interaction*, 2005, pp. 1423–1426.

[19] G. Jacucci, A. Spagnolli, L. Gamberini, and P. Monti, “Designing Effective Feedback of Electricity Consumption for Mobile User Interfaces,” *PsychNology*, vol. 7, no. 3, pp. 265 – 289, 2009.

A Novel Mobile Communication for the Future Internet

Sung-Moon Shin, Min-Taig Kim and Dae-Sik Kim
Electronics and Telecommunications Research Institute
Daejeon, Korea
smshin@etri.re.kr, mtkim@etri.re.kr, dskim@etri.re.kr,

Abstract — **In this paper, Organic Mobile Communication (OMC) is defined as a Mobile Communication (MC) system operating organically as a body. As smart phones, for example, the I-phone, become more popular, Internet services are becoming more intelligent and diversified. Internet traffic is also expected to increase explosively, with much variation. In the future Internet, MC is expected to play a key role. Therefore, the future MC needs to have more capabilities in capacity, function, etc. Especially, considering traffic requirements and limited radio resources, flexible operation of MC will be more important. OMC is an alternative to meet these requirements. In this paper, the architecture of OMC is analyzed with the service requirements from the system design point of view. OMC consists of the handsets, Base Stations (BS's) and the Core Networks (CN's) of the present MC system. However, the routing in the air is quite different. In OMC, all the wireless equipment, such as handsets and the BS's, are regarded as nodes, of which the states are ABLE or UNABLE. The routes among nodes are constituted by the ABLE nodes and are changed dynamically. An optimal route between two nodes is determined with aids of the Radio Routing Agent (RRA) of the CN. With OMC, we can increase the system capacity significantly and thus satisfy the future Internet requirements.**

Keywords – mobile; communication service; Internet; organic communication.

I. INTRODUCTION

As more convenient telecommunication services are required, the need for the Internet and MC also increases. Particularly, ubiquitous and intelligent services including convergence are very important for the future of MC [1]. Mobile Internet (MI) services are main services in the future of MC. The MI services require various levels of Quality of Service (QoS). The MC systems supporting these MI services are also varied. A user in the future will adopt the various MC systems. Direct radio paths will be also needed to support MI services efficiently. Considering the increased traffic in the future due to the ever-rising popularity of intelligent phones, it is important to improve the radio resource efficiency. Although the Universal Access (UA) with cognitive radio is proposed to improve the radio resource efficiency [2]-[3], it is difficult for UA alone to meet future MI requirements.

The Future MC System (FMCS) is expected to consist of several types of the MC systems, such as LTE, WiMax, WiFi, etc. The ad-hoc network [4] may also be a part of the FMCS. In the future, MC companies will serve their customers well with the required MI services, regardless of the supporting system. Also, as the smart phone becomes more intelligent, an MC system needs to be more intelligent. The FMCS needs to have the capability to support future MC users efficiently with MI services.

In this paper, a novel FMCS is proposed as an alternative to meet future MI requirements. In the proposed FMCS, each piece of wireless equipment, such as a handset or BS, is regarded as a node. Every node is available or unavailable. The route between two nodes consists of available nodes. The RRA maintains the status data about all of the nodes and the handsets. If a call is attempted, the RRA determines the optimal route for the call and informs the caller. The assigned route is changed dynamically, like an organism. With the proposed **Organic Mobile Communication** (OMC), radio resource efficiency is significantly increased by the flexible use of the resources.

This paper is organized as follows. In Section II, the life style of a future user and future MC services are discussed briefly. Section III presents an example of a part of the OMC architecture and its concept. A call processing procedure is also illustrated. Section IV is concerned with the wireless and mobile technologies required for the OMC. Finally, Section V concludes this paper.

II. MOBILE COMMUNICATION SERVICES

To extract the requirements for future MC services, , we briefly consider some aspects of life in the future . Compared to the present, the following kinds of people and other rising social phenomena are expected to be more common in the future.

- Lonely single
- Older women socializing with younger men
- Mobile couples from the weekend couples.
- Retired worker
- Extreme commuter
- Person who sleeps less
- DIY (do-it-yourself) doctors
- Learning requirements for teenagers
- Knitting youngster
- Adult video game
- Home schooling
- Easily and fast transportation

Considering the above, future life will be more personalized and more intelligent. In future society, human's existing problems will be solved. Such biological problems as illness, death, and pain will be relieved. Human and cyber space will be connected with special equipment to the human body. Artificial machines, such as artificial material, artificial software, artificial infra, and robot replace human's brain work as well as

ISBN: 978-1-61208-152-6

human's physical work. The care economy will rise jointly with the artificial society, since the care for human themselves in terms of mental and physical activity will be more important.

In the future, the advent of super brains will open the Artificial Intelligent (AI) society and thus an ideal society will be come into being. Humans will be able to reach self-realization through proxy experience, services, travel, games, leisure, stories, poems, novels, plays, movies, etc. Nano-technology will also be popularized and thus equipment will be minuscule. In addition, a large portion of human manual labor and mental work will be done by small robots and AI. In industry, humans will cooperate with robots. The trends of future life can be summarized by the terms 'carbon-reducer', 'sweet-interlude', 'communion-machine' and 'big-brain'.

To cope with the trend, MC is expected to play a key role with MI. Considering future life styles and trends, future MC services will be 'green-mobile', 'relaxation-mobile', 'communion-mobile', and 'brain-mobile'. The All Things on Network (ATON) including the machine to machine (M2M) is also a key service for the FMCS, with location-based-service (LBS). Therefore, future MC services are summarized as group communication (conference-call), community services (cyber-society), multimedia-query (specific-conversation), value-added data goods (including sight, auditory, smell, touch, etc.), and green transportation. Fig.1 shows an example of future MC services.

III. OMC ARCHITECTURE

The major service of the FMCS is MI service with M2M service. Since MI service and M2M service are provided at the present time, the continuous support of MI service and M2M service provided by existing systems, such as LTE, WiMax, and WiFi, is very important. Therefore, the existing systems, including the Intelligent Transportation System (ITS) and the Ubiquitous Sensor Network (USN), need to be adopted efficiently. Considering these, the FMCS is expected to consist of the existing systems as well as the new FMCS components. Furthermore, considering limited radio resources, the FMCS is required to increase the frequency-use efficiency significantly.

To meet the requirements, OMC is proposed. Fig. 2 shows the concept of OMC. OMC is a novel FMCS characterized by key words like 'informative', 'green', 'cognitive', 'self-organized', etc. Existing MC systems and the new FMCS components constitute the proposed OMC. In this OMC, all network components and radio resources are organically combined without any human control to provide the services optimally matched with the users' needs. OMC evolves by itself to adapt dynamically to the varying environments. Hence, with OMC, we can support future MI services as well as conventional MC services efficiently and can relieve the limitations of radio resources.

Fig. 3 shows an example of an MC part of OMC architecture. In Fig. 3, terminal A, base station A, and the core network A constitute the system A. System B consists of terminal B, base station B and, core network B. OMC may include other systems available in the operating company. In OMC, all wireless equipment, such as handsets and BSs, are regarded as nodes. The state of a node is described as either available (ABLE) or unavailable (UNABLE). Direct communication between any two nodes is possible, and the route consists of the ABLE nodes. The optimal route between two nodes is determined with the aid of the RRA of the CN A and is changed dynamically by the environment. In the figure, system A is assumed to be the main operating system. The RRA maintains all the status data about all the nodes of the handsets and the BSs. Hence, OMC is self-organized and operates organically.

Fig. 4 shows an example of a call processsing flow. If subscriber A makes a call, this message is transferred to the call processsor (CP) A of the CN A. If the CP A receives the call attempt message, the CP A requests the optimal route to the RRA. The RRA determines the optimal route for the call and informs the CP A. The CP A transfers the call attempt message to the CP B of the called subscriber. If there is no problem in the received response, the CP A informs the caller of the optimal route. With the informed route, the caller communicates with the callee. During the call, the route may be changed by the RRA like a handoff.

Fig. 5 shows the uplink channel architecture proposed for OMC. The basic channel architecture is the same as for LTE. In Fig. 5, a new channel, the Dedicated Access CHannel (DACH) is included to improve access efficiency. During the call processing procedure, the identification number for the DACH is informed by the network.

IV. WIRELESS AND MOBILE TECHNOLOGIES

OMC is proposed to efficiently support future MC services with MI services. Considering future MC requirements, radio resource efficiency is a key parameter for the system. Although a number of technologies are proposed to improve radio resource efficiency, it would be difficult to meet the requirements of MC in the future with the proposed technologies. However, the technologies could still utilized for OMC if they were improved, and here they are introduced.

- Multiple-Input Multiple-Output (MIMO). MIMO shares the communication resources by means of distributed transmission and signal processsing. Neighboring infra-structure stations as BSs or Relay Stations (RSs) are for utilization.
- CoOrdinated Multi-Point (COMP). COMP transmission is used with coordinated scheduling/beam-forming and joint transmission.
- Cooperative Processing (CP). CP overcomes the limits on spectral efficiency imposed by inter-cell interference. Between BSs and RSs, as well as among RSs, it extends the coverage and capacity of point-to-multi-point links.
- Multi-Hop Relay (MHR). A relay path is a concatenation of consecutive relay links between the source and the designated access relay station. The

ISBN: 978-1-61208-152-6

relay normally works in half-duplex mode. Three schemes, amplifying and forwarding (AF), the decoding and forwarding (DF), and estimating and forwarding (EF), are used.

- Cooperative relaying is a source node multi-casting a message to a number of cooperative relays which in turn resend a processed version to the intended destination node. The destination node combines the signals received from the relays and recovers the source signal
- Cognitive Radio (CR) allows universal access to the less-loaded-station. The traffic load is distributed among the MC systems available.
- Femtocell is an MC system for a small area, such as a home or the SOHO. In a femtocell, a compact BS is connected to the operator's network via the Internet. The operations of femtocells are controlled by the Self Organizing Network (SON).
- SON is a set of use-cases covering all the fields of network operation from network planning to maintenance activities. The self-optimization is also done in the SON. Plug and play technology is applied to the compact BSs to achieve the SON functions in conjunction with the CN.

V. CONCLUSION

Future communication services are expected to be more intelligent and convenient. MI services are the main services of the FMCS. Hence, a large capacity, a number of functions, and various levels of QoS are required for the FMCS. Although many wireless/mobile technologies, such as the MIMO, the CR, etc. are proposed to overcome the limited radio resources, it is difficult to meet the FMCS requirements only with the technologies. The proposed OMC in this paper is an alternative to meet future MC requirements.

OMC consists of handsets, BSs, and CNs. Unlike in current MC systems, direct communications among handsets are possible in OMC. In OMC, the different types of wireless equipment are regarded as nodes. The optimal route between two nodes is constituted by available nodes and is assigned with the aid of the RRA of the CN. The RRA maintains the status data about all nodes. The assigned route is changed dynamically, as if it were an organism. With the proposed OMC, radio resource efficiency is significantly increased by flexible use of resources.

ACKNOWLEDGMENT

This research was supported by the KCC (Korea Communications Commission), Korea, under the ETRI R&D program supervised by the KCA (Korea Communications Agency)" (KCA-2011-08921-05001].

REFERENCES

[1] Vekasalo, H. et al., “Analysis of users and non-users of smartphone applications,” Telematics and Informatics., vol. 27, pp. 242-255, Aug. 2010.

[2] Sherman, M, et al., “IEEE Standards Supporting Cognitive Radio and Networks, Dynamic Spectrum Access”, IEEE Comm. Magazine, vol.46, pp. 108-116, July 2008.

[3] Prasad, R.V., et al., “Cognitive functionality in next generation wireless networks: standardization efforts”, IEEE Comm. Magazine, vol. 46, pp72-78, Apr. 2008.

[4] Sommer, C. et al., “On the feasibility of UMTS-based Traffic Information Systems,” Ad Hoc Networks, vol 8, pp. 506-517, July 2010.

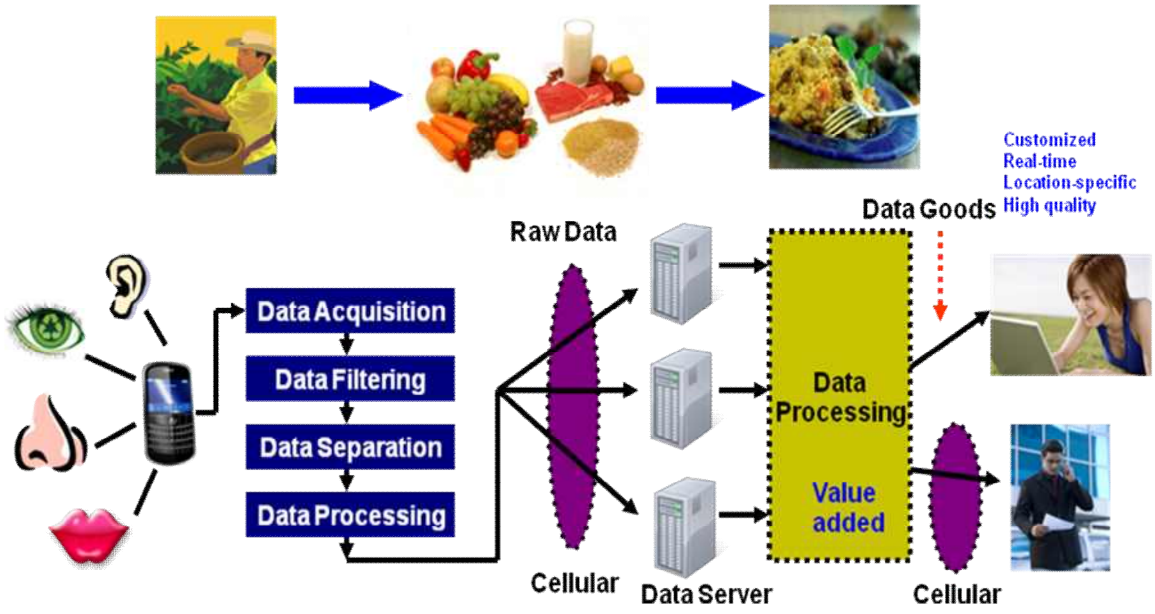


Figure 1. An example of the future MC services

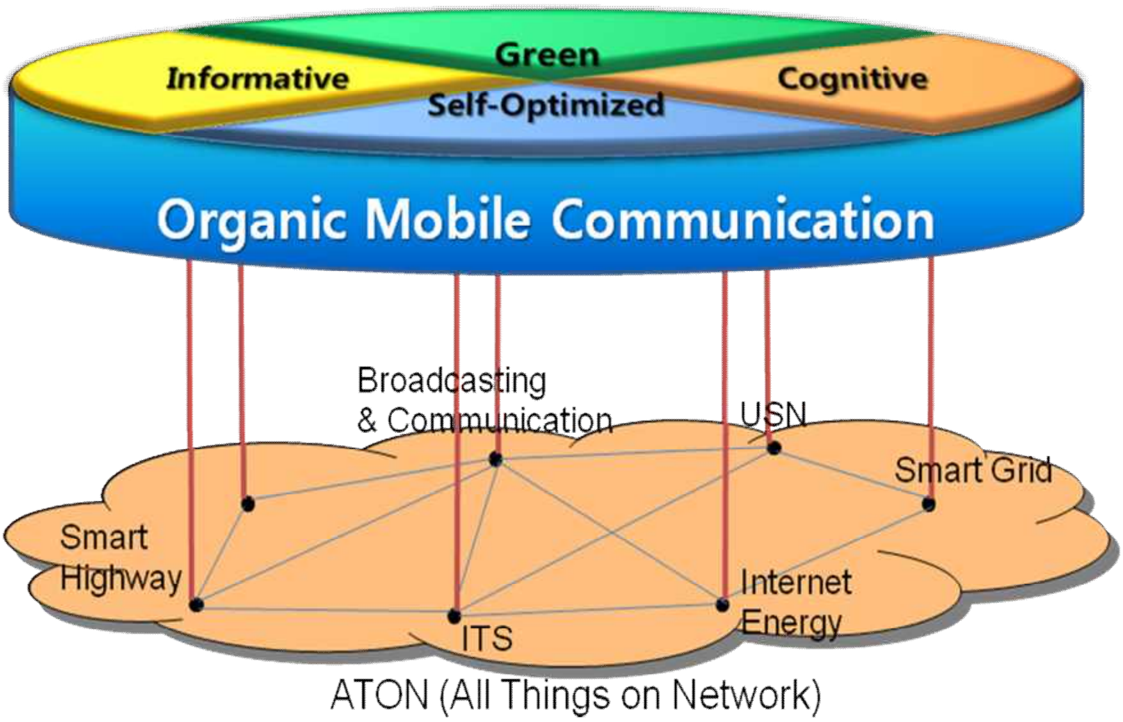


Figure 2. The concept for the organic mobile communication

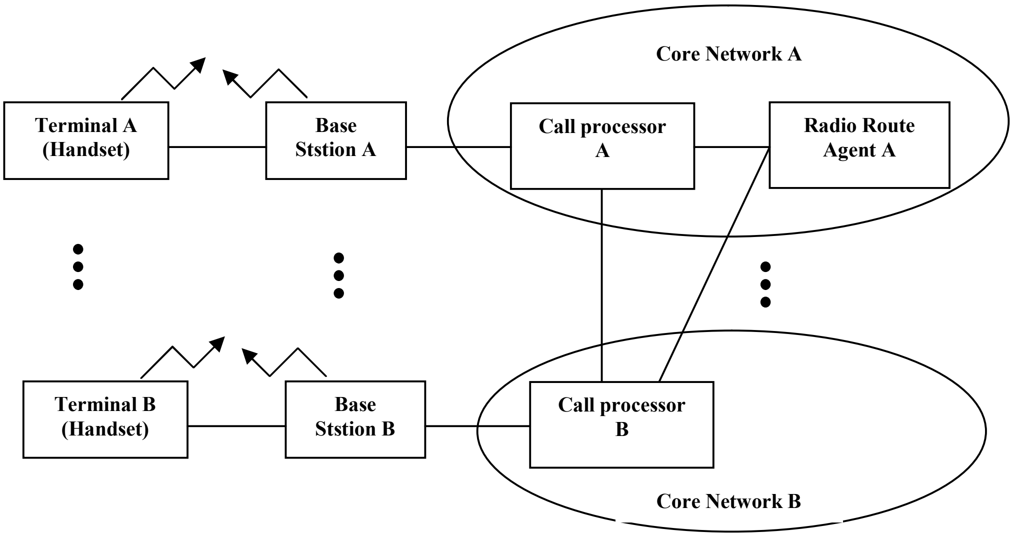


Figure 3. An example of a MC part of the OMC architecture

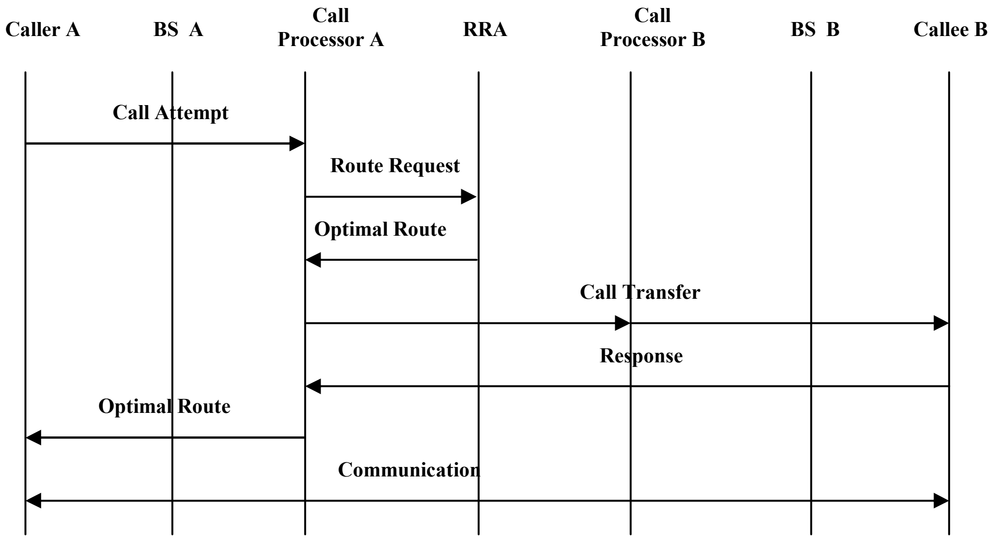


Figure 4. An example of a call processing flow of OMC.

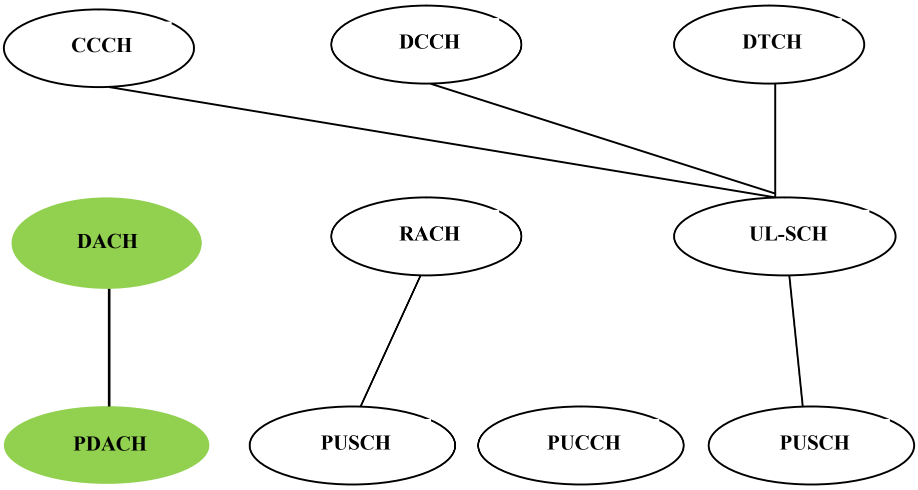


Figure 5. The uplink channel architecture for the OMC.

ISBN: 978-1-61208-152-6

A Domain Ontology for Designing Management Services

Ingo Pansa[1], Matthias Reichle[2], Christoph Leist[2], Sebastian Abeck[1]
Cooperation & Management
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
[1]{pansa, abeck}@kit.edu
[2]{matthias.reichle, christoph.leist}@student.kit.edu

Abstract— **Designing management systems based on service-oriented principles is a pragmatic approach to handle the challenges that distributed management is faced today. In order to conform to service-oriented principles, the elements of the management systems architecture – the management services – have to be designed along domain-specific concepts. Thus, modeling the domain IT Management becomes evident within service-oriented development processes. Considering existing approaches, domain modeling is not addressed explicitly, thus hampering the construction of management systems based on service-oriented principles. In this paper, we propose an ontology for the specification of the domain IT Management and present a refined development approach that enables the application of the presented ontology. The application of the ontology is demonstrated by designing management services for a standardized Incident Management Process.**

Keywords - management service; ontology; domain model

I. INTRODUCTION

With the adoption of web-based dynamic IT services by the business, management of distributed IT infrastructures becomes an evident part of a service provider's daily operations [22, 32]. Defining, designing and implementing management processes within the providers organization is therefore necessary. Automating these management processes is essential, as both reaction times can be limited and recurring errors by technical personnel involved in these processes are avoided [34]. Constructing a service-oriented solution built upon loosely coupled, process-oriented and reusable management services is desired. However, the challenges in realizing this are numerous [17, 21, 22, 23, 31]. From the perspective of an IT infrastructure operator, IT services are often created using a couple of different computing systems running different applications, interconnected by different networking technologies. In real life scenarios, often some hundred different management tools are utilized. Often these tools do not offer any standardized interfaces hampering the automation of management processes [24, 25].

In order to fully utilize the principles of service-oriented computing within the domain of IT Management, a clearly defined development process focusing the constructing of reusable management services is needed. This development approach has to consider both IT management standards and aspects of how to design service-oriented software solutions likewise. Although initial work discussing the application of service-orientation to construct management systems exists [6, 7, 21, 24, 25], little has been done to tackle this challenge on a conceptual level by explicitly focusing on the reusability and process orientation of flexible management systems.

In this paper, we deliver contributions to the addressed problems [2, 3] in that we clearly specify different aspects of models for designing reusable management services. The refined models are based on OWL ontology [4], enabling both the definition of a meta model and the application of the meta model to different management areas [36, 37]. Focusing an ontology-based definition of domain models not only supports the construction of reference models that can be shared among the scientific community [1, 9, 11, 14] but also the construction of concrete management tool support by the tool vendors [36, 37].

The remaining parts of this paper are structured as follows: Section 2 introduces related work and provides the background for applying ontological engineering for designing service-oriented systems. In Section 3, we introduce a refined development process for service-oriented design embedding an ontology-based meta model. Section 4 presents the contribution of this paper: we specify the conceptual meta model using an OWL ontology based on the refined definition of the necessary abstract syntax. In Section 5, we demonstrate the application of a reference model for designing management services for Incident Management that are both reusable and aligned to the management processes that they support. Finally, Section 6 concludes this paper and gives an overview of the work that is currently being done within our research group.

II. BACKGROUND AND RELATED WORK

Applying domain-driven techniques for designing management services that address the challenges discussed in [21, 22, 23, 32, 34], a revision of the approaches contributing to service-oriented management systems [6, 7, 24, 25, 42] becomes necessary. As we currently observe a shift towards web-based usage of dynamic IT Services ("Cloud Computing"), this holds even more. Standards such as ISO/IEC20000-1:2005 [19] only serve as a starting point, but lack of a foundation that is based on formalism thus impeding the efficient development of flexible management systems.

As Erl [40] states, the analysis phase is the most important step in the service-oriented software development process towards well-designed services. Having a proper tool support by utilizing according modeling languages greatly influences the quality of the resulting analysis artifacts, as models have several advantages compared to informally defined analysis artifacts [12]. Following such structured development processes, analyzing, defining and modeling the domain for the desired information system serves as a starting point [8, 15]. Explicitly considering a certain domain greatly increases the possibility of engineered components to be reused [39].

As stated in [10], development teams tend to suffer from "UML fever" thus creating models that are too fine grained. While this is certainly true in general, considering analysis phases of typical development processes, both the support of modeling languages and the ability to describe the considered aspects in an intuitive way is desired likewise. Several recent works [1, 16] propose the use of ontology [18] for domain analysis mainly as a starting point leading to a meta model and later to a domain model [20, 33, 35]. Other authors [9, 13] even go one step further and use ontology as meta model or domain model itself. To unify both UML and ontology-based modeling within analysis phases, the OMG currently considers the definition of an overall UML-based meta model for ontologies, thus allowing to use several concrete syntaxes for ontology definition [29].

In our approach, this is exactly what we intend to do by utilizing the Web Ontology Language OWL [4].

III. DOMAIN-DRIVEN DESIGN

In order to be able to organize the different steps of a complex service-oriented software development project, different development process models have been proposed in the past (e.g., [5]). Their goals range from easy-to-adopt agile process to complex frameworks considering legal or cultural backgrounds of the involved parties. Focusing domain-driven design, several approaches have been discussed. While it is not our intend to present yet another development method, a basic understanding of the different steps required within service-oriented software development is required in order to comprehend and adopt the proposed domain ontology.

A. Overall development process

On a very abstracted level, software development is structured along different phases: based on requirements elicitation [41], the (possibly) informally defined requirements are analyzed and modeled using formally defined modeling languages. Two concerns are primarily considered within this analysis phase: structural analysis and dynamic analysis. While structural analysis deals with the definition of the domains, borders, stakeholders or objects, the dynamic analysis focuses the interaction of the elements that are identified within structural analysis. For both structural and dynamic analysis exists a couple of different models and modeling techniques, for which UML-based approaches can be regarded as the ones mostly preferred by software engineers. Due to the artifacts that are produced by the overall development process, the considered analysis phase is refined as service-oriented analysis. Different approaches are proposed to define necessary steps and development activities within service-oriented analysis [40].

Following service-oriented analysis, service-oriented design focuses on the definition of prescriptive models that clearly define the semantics of the to-be-implemented artifacts in means of the underlying platform concept. As a service-oriented software solution is desired, the underlying platform concept is bound to the principles of service-orientation (such as loose coupling, message-based communication, clear business relation).

B. Models

Based on the hitherto discussed development process, several models serve as a foundation to support the different development tasks. As we focus the overall analysis and design phases of the development process, we suggest no further assumptions on how to model the different artifacts that are produced during requirements elicitation. Typically, UML Use Cases are used to sketch up the desired functionality on a coarse granular level, supporting an understanding of both customers and business process analysts.

Focusing service-oriented analysis and service-oriented design, the central artifacts that are produced within these phases are domain models (analysis phase), service candidate models (according to Erl [40] in analysis phase) and service design models. To capture the overall choreography that different involved partners in a complex business scenario inhabit, business process models are used.

Due to the different semantics of each of the concrete models, different modeling languages are used. In Fig. 1, an overview of the different models and their interrelations is presented.

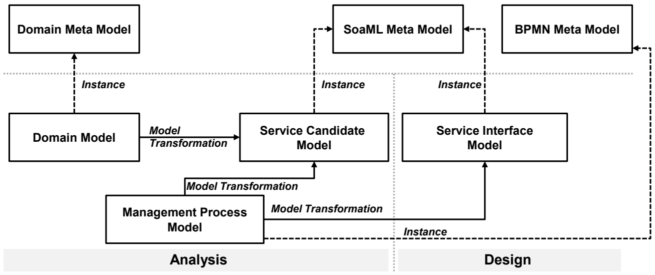


Figure 1. Different models in analysis and design phases

Both domain models and the meta model are defined using the Web Ontology Language (OWL) [4]. Although OWL originally was intended to be useful for semantically enriched resources in the World Wide Web, lately published work highlights the advantages of using both OWL to define project-specific analysis models and project-independent meta models. Business process models capture dynamic aspects of the structural elements that are modeled using domain models, wherefore both domain models and business process models are used to derive service candidate models. Business process models can be defined using different languages from which one of the most accepted is the Business Process Modeling Language (BPMN), published by the OMG [28]. Considering the definition of service candidate or service interface models, we propose to utilize SoaML [30]. This upcoming standard for modeling service-oriented systems published by the OMG enables to define several aspects of service-oriented systems. One of the most interesting in our opinion is the concept of a service candidate, that defines required but not yet fully specified service capabilities. Based on the definition of service candidates, service interfaces can be engineered focusing the platform-specific requirements that service-oriented software systems are build upon.

C. *Model Transformations*

As domain models capture the structural aspects of the software system that is to be designed, using domain models to define the dynamic aspects prevents the involved stakeholders of defining the different elements with divergent semantics. Focusing on domain models being defined using OWL and process models being defined using BPMN, a simple transformation scheme can be defined as shown in Table 1.

TABLE I. DOMAIN META MODEL AND BUSINESS PROCESS META MODEL

Domain meta model element	Process meta model element
Management Area	Pool
Management Participant	Lane
Management Entity	Data Object
Management Basic Activity	Task
Management Composed Activity	Sub-Process

Although the transformation is defined informally, it proved that the process models we derived on the modeled OWL ontologies were much more intuitive to understand as they contained exactly the identified management participants, their executed activities and their required management entities.

IV. A DOMAIN ONTOLOGY

In addition to previous work we published so far [2, 3], an approach based on formal domain models has several advantages. Using OWL to define such formal semantics, in this section we outline the definition of a domain ontology based on our comprehensive abstract syntax.

A. *Abstract syntax*

The first step towards an ontology based on the requirements defined in ISO/IEC20000-1:2005 [19] is to gather the relevant vocabulary. In our previous work [2, 3] we derived the key concepts of the IT Management domain from the specification which are elaborated further with this paper. As a result, we identified certain concepts that can be used to model the IT Management domain in a functional centric way, providing the basis for the design of reusable and process oriented management services.

The central element of the domain is the *Management Area*, which represents one of the thirteen management processes like Incident Management or Change Management described in [19]. It holds all other entities related to a particular management process. Every Management Area consists of so called *Management Activity* elements which are used to denote a concrete activity performed to accomplish management goals. There are two specializations, namely *Basic Management Activity* and *Composed Management Activity*. A Basic Management Activity cannot be drilled down any further; it is atomic, whereas a Composed Management Activity is used to embody complex Workflows which consist of one or more Management Activities. In addition to the concepts of ISO/IEC20000-1:2005, we propose the modeling of *Management Capabilities*. A Management Capability is an abstract element that describes a certain capability which is used in Management Activities to fulfill their tasks. They can be refined to *Provided Management Capability* or *Required Management Capability*, depending on whether it is provided by a service e.g., a management tool or required by an activity, thus playing a major role in integrating existing functionality. To represent all necessary information required by a management process, we use *Management Entities*. Using the *Management Participant* concept, we model all actors involved in activities that belong to a certain Management Area. The last type of relevant information that needs to be modeled is a flexible way to describe requirements in regard of structure or handling of other elements in the domain, therefore we use *Management Policies*. One specialization of Management Policy is a *Management Entity Structure Policy* which specifies the structure an entity has to fulfill.

After identifying these basic concepts of IT Management, we have a solid foundation on which the IT Management ontology can be built on.

B. *Ontology of the meta model*

Using the OWL Web Ontology Language, we aim at a more formal specification of the key concepts introduced before. The specification heavily relies on the description of relations between OWL classes and restrictions regarding the validity of such relations between those elements. Furthermore, we define necessary conditions that elements of a specific class have to fulfill. Meta model elements (i.e. OWL classes) are not only restricted through abstract super classes but also inheritance of restrictions according to their parent class in the hierarchy of the OWL classes (see Fig. 2). Hereby, we obtain a framework describing how management areas can be modeled in a way that conforms to the meta model.

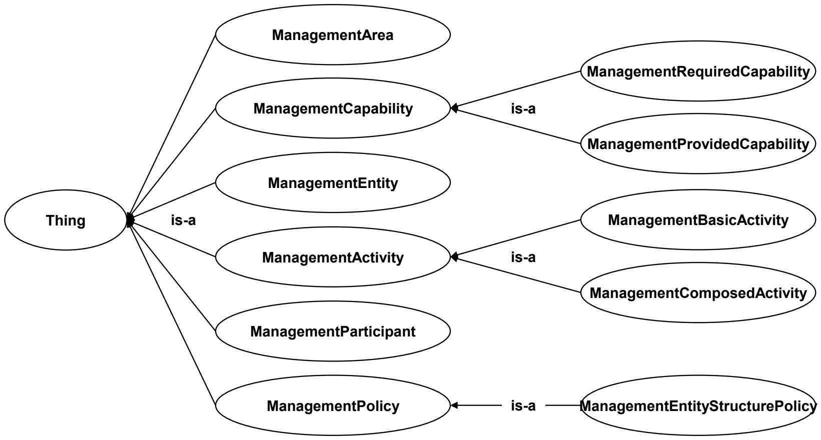


Figure 2. Taxonomy of IT Management

The associations between model elements are realized with OWL Object Properties. Based on [19], we identified six relations between model elements, where each one has an inverse. The relations below are depicted in the following style: *relation (~inverse)*.

Analogous to the central element in the meta model (Management Area) there is a corresponding central relation between the elements. The *contains (~isPartOf)* Object Property describes that a Management Area contains certain other elements, specifically those are Management Participant, Management Entity, Management Policy, Management Activity and Management Capability. To model information about various actors related to activities, we introduce *participatesIn (~hasParticipant)*. The fact that activities rely on capabilities to reach management goals is taken into account by the *requires (~isRequiredBy)* relation. As stated above, Management Activities may make use of Management Entities to gather or to store information. Therefore we use an Object Property called *operatesOn (~isOperatedBy)*. The entity itself is described in detail using a Management Entity Structure Policy that is bound to the Management Entity by *defines (~isDefinedBy)*. The sixth pair of relations models the composition of Management Activities. To represent the concept of composition, we use *isComposedOf (~isUsedBy)*.

```
(contains some ManagementActivity)
 and (contains some ManagementParticipant)
 and (contains only
     (ManagementActivity
     or ManagementCapability
     or ManagementEntity
     or ManagementParticipant
     or ManagementPolicy))
```

Figure 3. OWL Restrictions for Management Area class

The relations introduced in the paragraph above in conjunction with the key concepts identified earlier are now used to further specify the demands we make on domain models of IT Management. This is achieved through heavy use of OWL Restrictions, which can be seen in the excerpts of our OWL definition above (Fig. 3).

The Management Area element may hold the elements depicted earlier but to be a valid Management Area in our understanding it has to contain at least one Management Activity as well as a corresponding Management Participant. This results in the following OWL Restrictions for Management Area.

Every individual inside or every class inheriting of the Management Area class thus has to satisfy those requirements. To specify the nature of Management Activities, we restricted the members of the Management Activity class as shown in Figure 4.

```
(isPartOf only ManagementArea)
 and (isPartOf exactly 1 ManagementArea)
 and (hasParticipant some
ManagementParticipant)
 and (hasParticipant only
ManagementParticipant)
 and (operatesOn some ManagementEntity)
 and (operatesOn only ManagementEntity)
 and (requires some ManagementCapability)
 and (requires only ManagementCapability)
 and (ManagementBasicActivity or
ManagementComposedActivity)
 and (isUsedBy only ManagementComposedActivity)
```

Figure 4. OWL Restrictions for Management Activity class

The code presented in Fig. 3 and Fig. 4 reflects the fact that each Management Activity belongs to exactly one Management Area and it has to have at least one participant. Furthermore, we specified the necessity of a corresponding Management Entity as well as one or more Management Capabilities that are used by it. The last two statements say that each activity is either a basic or a composed activity and if it's composed, it can only be used by another Management Composed Activity.

All the other elements of the IT Management ontology are formalized in the same manner, resulting in a rather complex description of concepts and their interrelations in the IT Management domain as seen in Fig. 5.

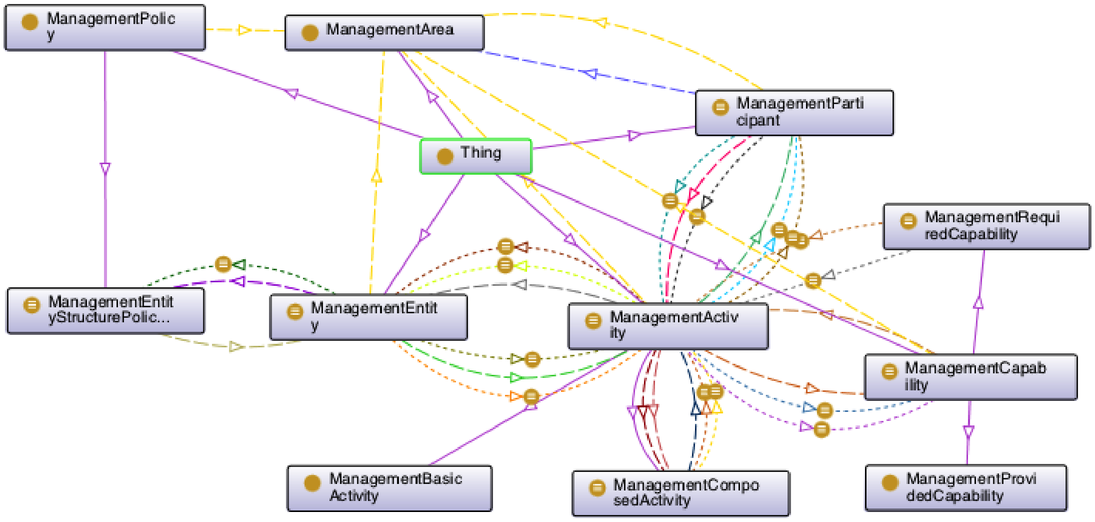


Figure 5. Domain ontology for IT Management

Based on this meta model, we now intend to describe the thirteen management processes mentioned in ISO/IEC20000-1:2005.

V. Designing Management Services for Incident Management

The following application example is divided into two parts: Part 1 deals with the definition of the domain ontology for Incident Management while part 2 demonstrates the design of concrete management services based on the proposed domain ontology.

A. Ontology for Incident Management

The domain ontology for the specific Management Areas is constructed by extending the OWL ontology of the meta model following some basic rules, the most important one being not to introduce classes directly under the Thing root class. Hereby, all classes of the domain model are restricted by the definitions of the meta model elements.

As a preparatory step, we need to identify the relevant elements that are needed to model the desired management process, which is described in detail in our previous works [2, 3]. As an example, we identified the elements shown in Table 2 for the Incident Management Process.

TABLE II. Elements for Incident Management ontology

Meta Model Element	Domain Model Element
Management Area	Incident Management
Management Participant	Incident Manager, ServiceDeskEmployee, Specialist
Management Entity	Entity dedicated to Incident Management: Incident Record Further entitites: Known Error Record, Workaround Record, Configuration Management Database Record (CMDB Record)
Management Activity	Record Incident, Determine Business Impact, Prioritize Incident, Classify Incident, Escalate Incident, Resolve Incident, Inform Customer, Create Incident Record, Update Incident Record
Management Policy	Incident Record Structure Policy

The aforementioned elements now can be specified through further restrictions in line with the ones being specified at the meta model layer of the ontology. This leads to a well described model of the Incident Management Process according to ISO/IEC20000-1:2005 and valid in respect of the meta model, as seen e.g., in the fragment of our ontology below (Fig. 6), covering details of the Management Area Incident Management. These restrictions are added to the ones inherited through the hierarchy which places Incident Management beneath Management Area.

```
(contains some CreateIncidentRecord)
 and (contains some
CreateIncidentRecordCapability)
 and (contains some IncidentRecord)
 and (contains some
IncidentRecordStructurePolicy)
 and (contains some RecordIncident)
 and (contains some ServiceDeskEmployee)
```

Figure 6. OWL Restrictions for Incident Management area

For instance, we can see that Incident Management contains an element called Create Incident Record that actually is a Management Activity belonging to the said Management Area. On the other hand the definition of Create Incident Record contains the inverse of that information as well as other relations to further elements.

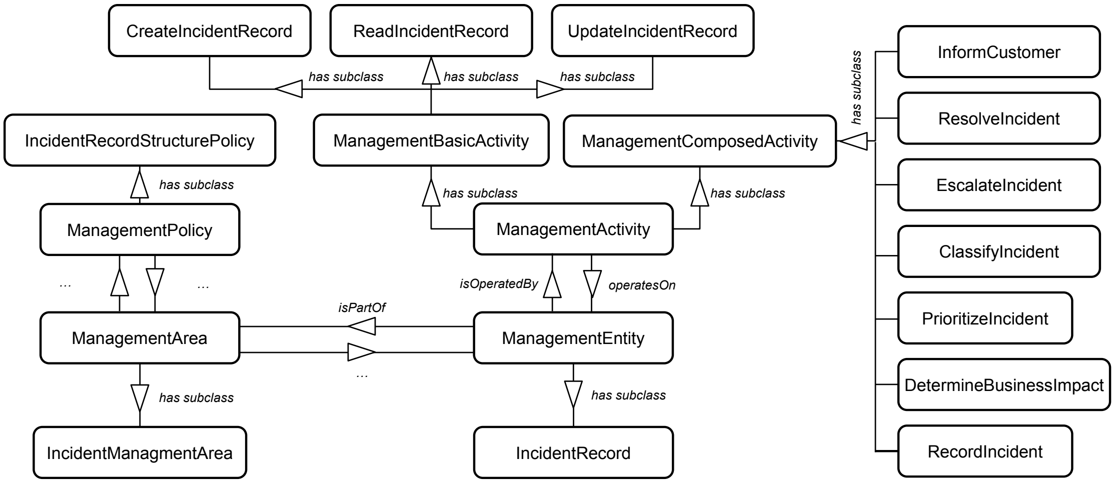


Figure 7. Domain ontology for Incident Management

By describing each element of Table 2 in the way seen above, we achieve a domain ontology for Incident Management forming a reference model according to Fig. 5. Fig. 7 shows a graphical excerpt of the ontology for Incident Management.

B. Incident Management Service Design

With the definition of ontology for Incident Management, the design of management services supporting automated management processes can be based on a solid foundation that transforms functional requirements to executable code. Following the overall development process as presented in Section 3, in this section we briefly give an overview of how to leverage the benefits of ontology-based domain-driven software development focusing management services.

Following the rules presented in [2] and taking into account the domain-knowledge formally specified in the OWL ontology, we are now able to construct management services in a comprehensible and repeatable way using SoaML [30]. One of the rules states that for every Management Entity a SoaML Message Type object should be created. Considering the Incident Management Process, we identified only one element residing directly beneath the Management Entity class in the OWL-hierarchy. Therefore we model a SoaML Message Type with the name of Incident Record and the structure defined in the Management Entity Structure Policy referenced through the isDefinedBy Object Property. Another principle we propose is to introduce a SoaML Capability that groups all Management Capabilities of the domain ontology that handle a single Management Entity. This results in a SoaML Capability named Incident Record Service which contains the operations according to the Management Capabilities of the OWL ontology, namely Create Incident Record, Read Incident Record and Update Incident Record. By applying some more rules as the ones we are able to present within the scope of this work, we are able to identify the most important SoaML model elements based on elements of our OWL ontology and their position inside the hierarchy, which places every element of the domain below one specific element of the meta model layer.

The SoaML results of the procedure described above can be seen in Fig. 8 showing SoaML Capabilities and Service Interfaces.

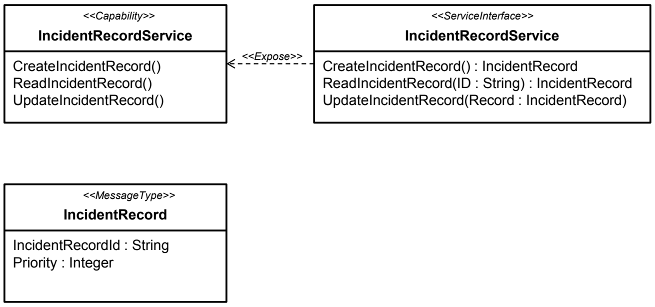


Figure 8. SoaML Capability and Service Interface reflecting domain concepts

Through further steps the SoaML Service Interfaces can be refined and finally realized by implementing Web services using technologies like WSDL [38] or its semantic annotated sibling SAWSDL [26] (see Fig. 9).

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="IncidentRecordService"
[...]
xmlns:sawsdl="http://www.w3.org/ns/sawsdl">
[...]
<xsd:complexType name="IncidentRecord"
      sawsdl:modelReference="http://domp.cm-
tm.uka.de/ontologies/ITSMOntology#IncidentRecor
d">
   <xsd:sequence>
      <xsd:element name="IncidentRecordId"
       type="xsd:string"></xsd:element>
      <xsd:element name="Priority"
       type="xsd:int"></xsd:element>
   </xsd:sequence>
</xsd:complexType>
[...]
```

Figure 9. SAWSDL excerpt showing references to the domain ontology

As the application example demonstrates, the semantics of the modeled domain artifacts can be preserved into the subsequent design steps. This finally helps to close the gap between analysis and design phases thus lowering development efforts in round trip engineering.

VI. Conclusion

For tackling the complexity that distributed management is faced to today, the automation of management processes is evident. Building management systems upon loosely coupled management services strongly supports this approach. However, several aspects have to be considered, as service-oriented design and analysis is a creative software development issue in which several different stakeholders are involved. For being able to model requirements and support the development process using model-based transformation techniques, analyzing and understanding the domain is essential. This paper addresses this situation and delivers several contributions.

In Section 3, a refined development process for service-oriented analysis and design was presented. The development process is generic in a way that concrete process models for software development are abstracted, but yet essential roles, stakeholders and artifacts are defined. The presented development process defined basic steps and activities that utilize ontological engineering and therefore served as a foundation for a concrete application of our approach in further scenarios or projects. Section 4 introduced a formalized meta model of the domain IT Management that served as a foundation for the definition of a reference model for Incident Management. The presented meta model focused problems that were previously addressed by our research group [2, 3]. Using OWL to define the domain ontology based on formal semantics not only allows to construct tool support that directly guides involved stakeholders during initial analysis or design activities but also to consider service design quality by the application of metrics suites. Section 5 demonstrated the application of the development process. We showed that utilizing ontological engineering supports the construction of management services that align with certain design principles, of which we mostly addressed in this paper the general requirement of a clear process-alignment of designed services.

Considering the experiences that we made within the development project, further work is necessary. Based on the development method, the OWL-based domain meta model and the OWL-based Incident Management reference model, introducing automated metrics applications of concrete service designs in regard of several service design quality aspects seems to be possible. For instance, semantically enriched service models would allow automated classification based on the modeled management area context, thus leading to possible better reusability if future requirements are slightly changing. Considering the evolution of the proposed ontology, both an ontology repository and corresponding tools are necessary.

References

[1] D. Gasevic, D. Djuric, and V. Devedzic, Model Driven Engineering and Ontology Development, Springer, Heidelberg, 2006.

[2] I. Pansa, F. Palmen, S. Abeck, K. Scheibenberber, “A Domain-driven Approach for Designing Management Services”, SERVICECOMPUTATION2010, Lisbon, 2010.

[3] I. Pansa, P. Walter, K. Scheibenberger, and S. Abeck, “Model-based Integration of Tools Supporting Automatable ITSM Processes”, IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), 2010, Page(s): 99 – 102.

[4] World Wide Web Consortium (W3C), Web Ontology Language (OWL), W3C Recommendation, 2004.

[5] A. Arsanjani, S. Gosh, A. Allam, T.Abdollah, S. Ganapathy, and K. Holley, „SOMA : A method for developing service oriented solutions,“ in IBM Systems Journal, Vol. 47 (3), pp. 377-396, 2008.

[6] G. Aschemann and P. Hasselmeyer, „A Loosely Coupled Federation of Distributed Management Services“, Journal of Network and Systems Management, Vol 9, No. 1, 2001.

[7] N. Anerousis, „An Architecture for Building Scalable, Web-Based Management Services“, Journal of Network and System Management, Vol. 7, No 1., 1999.

[8] G. Arango, Domain Analysis - From Art Form To Engineering Discipline, ACM SIGSOFT Software Engineering Notes, Volume 14, Issue 3, 1989.

[9] U. Aßmann, S. Zschaler, and G. Wagner, „Ontologies, Meta-models and the Model-Driven Paradigm“, Ontologies for Software Engineering and Software Technology (2006), pp. 249-273, 2006.

[10] A. E. Bell: Death by UML Fever, ACM Queue, Volume 2 Issue 1, March 2004, 2004.

[11] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, „What are Ontologies, and why do we need them?“, IEEE Intelligent Systems and their Applications, Vol. 14, Issue 1, pp. 20-26, 1999.

[12] G. Cernosek, and E. Naiburg, „The Value of Modeling,“ IBM Developer Works Whitepaper, 2004.

[13] V. Devedzic: Understanding Ontological Engineering, Communications of the ACM, Vol. 45 (4), pp. 136-144, 2002.

[14] R. de Almeida Falbo, G. Guizzardi, and K. C. Duarte, „An Ontological Approach to Domain Engineering“, Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE2002), 2002.

[15] X. Ferré, and S.Vegas, „An Evaluation of Domain Analysis Methods“, In 4th CAiSE/IFIP8.1 International Workshop in Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD99), 1999.

[16] T. Stahl, M. Völter, S. Efftinge, and A. Haase, Modellgetriebene Softwareentwicklung, dpunkt.verlag, 2007.

[17] S. D. Galup, R. Dattero, J. J. Quan, and S. Conger, „Information Technology Service Management: an emerging area for academic research and pedagogical development“, Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, pp.52, 2007.

[18] T. Gruber, „A translation approach to portable ontology specifications“, Knowledge Acquisition, Vol. 5, Issue 2, pp. 199-220, 1993.

[19] ISO/IEC20000-1:2005, Information technology – Service management – Part1: Specification, International Standards Organization (ISO), 2005.

[20] G. Kappel, E. Kapsammer, H. Kargl, G. Kramler, T. Reiter, W. Retschitzegger, W. Schwinger, and M. Wimmer, „Lifting Metamodels to Ontologies“, Lecture Notes in Computer Science, Volume 4199/2006, pp. 528-542, 2008.

[21] P. Kumar, „Web Services and IT Management“, ACM Queue, Volume 3 Issue 6, 2005.

[22] V. Machiraju, C. Bartolini and F. Casati, „Technologies for Business-Driven IT Management“, in Extending Web Services Technologies: the Use of Multi- Agent Approaches, Kluwer Academic, 2004.

[23] A. Moura, J. Sauve and C. Bartolini, „Research Challenges of Business-Driven IT Management“, 2nd IEEE/IFIP International Workshop on Business- Driven IT Management, pp.19-28, 2007.

[24] C. Mayerl, F. Tröscher, and S. Abeck, „Process-oriented Integration of Applications for a Service-oriented IT Management“, The First International Workshop on Business-Driven IT-Management, BDIM’06, pp. 29-36, 2006.

[25] C. Mayerl, T. Vogel and S. Abeck, „SOA-based Integration of IT Service Management Applications“, 2005 IEEE International Conference on Web Services (ICWS 2005), 2005.

[26] World Wide Web Consortium (W3C): Semantic Annotations for WSDL and XML Schema, W3C Recommendation, 2007.

[27] Organization for the Advancement of Structured Information Standards (OASIS), Reference Model for Service Oriented Architecture, Version 1.0, OASIS, August 2006.

[28] Object Management Group (OMG), Business Process Model and Notation BPMN, Version 2.0, OMG,Januar 2011.

[29] Object Management Group (OMG), Ontology Definition Metamodel (ODM). Version 1.0, OMG, Mai 2009.

[30] Object Management Group (OMG), Service-oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS), FTF Beta 1, OMG, April 2009.

[31] G. Pavlou, „On the Evolution of Management Approaches, Frameworks and Protocols: A Historical Perspective, Journal of Network and Systems Management“ Springer New York, Vol. 15, Issue 4, pp. 425- 445, 2007.

[32] J. Sauvé, A. Moura, M. Sampaio, J. Jornada and E. Radziuk, „An Introductory Overview and Survey of Business-Driven IT Management“, 1st IEEE/IFIP International Workshop on Business-driven IT Management (BDIM 2006), 2006.

[33] S. Staab, T. Walter, G. Gröner, and F. S. Parreiras, „Model Driven Engineering with Ontology Technologies“, Lecture Notes in Computer Science, Volume 6325/2010, pp. 62-98, 2010.

[34] V. Tosic, „The 5 C Challenges of Business-Driven IT Management and the 5 A Approaches to Addressing Them“, The First IEEE/IFIP International Workshop on Business-Driven IT Management, 2006.

[35] M.-N. Terrasse, M. Savonnet, E. Leclercq, T. Grison, and G. Becker, „Do We Need Metamodels AND Ontologies for Engineering Platforms?“, Proceedings of the 2006 international workshop on Global integrated model management, 2006.

[36] J. E. López De Vergara, V. A. Villagrá, J. Berrocal: Semantic Management: advantages of using an ontology-based management information meta-model, Proceedings of the HP Openview University Association Ninth Plenary Workshop (HP-OVUA'2002), 2002.

[37] J. E. López De Vergara, V. A. Villagrá, J. Asensio, J. Berrocal: Ontologies Giving Semantics to Network Management Models, IEEE Network, Vol. 17, pp. 15-21, 2003.

[38] World Wide Web Consortium (W3C), Web Service Description Language (WSDL) Version 2.0 Part1 Core Language, W3C Recommendation, 2007.

[39] J. Wang, J. Yu, P. Falcarin, Y. Han, and M. Morisio, „An Approach to Domain- Specific Reuse in Service-Oriented Environments“, Proceedings of the 10th international conference on Software Reuse: High Confidence Software Reuse in Large Systems, 2008.

[40] T. Erl: SOA, Principles of Service Design, Prentice Hall, 2008.

[41] B. Bruegge, and A. H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns and Java, Pearson Education, 2009.

[42] G. Tamm and R. Zarnekow, „Umsetzung eines ITIL-konformen IT-Service-Support auf der Grundlage von Web-Services“, Wirtschaftsinformtik 2005, pp. 647-666, 2005.

Designing Reusable Management Services

Ingo Pansa[1], Christoph Leist[2], Matthias Reichle[2], Sebastian Abeck[1]
Cooperation & Management
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
[1]{pansa, abeck}@kit.edu
[2]{christoph.leist, matthias.reichle}@student.kit.edu

Abstract— **Reusing functionality is one preferable requirement in today's engineering of distributed systems. Focusing IT Management systems as a key enabler to modern service-oriented systems, reusing management functionality can be achieved by applying the principles of service-orientation to support the construction of reusable management services. Thus, in order to construct these management services aligned with certain design quality, estimating the possible degree of reusability during analysis and design steps is required in order to support certain design decisions. Existing approaches targeting the design of management services do not take reusability into account explicitly, wherefore the proposed solutions seem to be hard to adopt if requirements to that system change. In this paper, an overall approach based on domain modeling is presented, supporting the design of management services by explicitly defined reusability metrics. The approach is exemplified by designing management services for a typical Incident Management scenario in which we outline the value of domain modeling for creating reusable design blueprints.**

Keywords - management service design; reusability; domain model

I. INTRODUCTION

As nowadays software systems grow in complexity, decoupling different parts of the systems is one of the most desired characteristics that system engineers follow. Different approaches have been proposed, starting with the early Client/Server-Architectures followed by CORBA [9] in the 90’s up to Web Service-based Architectures in the beginning of the new century. While all these approaches have major differences in how to structure the proposed architectures, they have some basic principles in common, of which the reusability of existing software artifacts seems to be one of the most important.

Focusing Service-oriented Architecture (SOA) [15], reusability of software artifacts is reflected in the existence of clearly defined service interfaces that hide details of the service implementation [27]. These service interfaces are expected to align with business process requirements thus supporting a basic reusability on a coarse granular level. Furthermore, standardized technologies such as Web Services or Universal Description, Discovery and Integration (UDDI) [33] are utilized to realize technical aspects of reusability.

While there seems to exist a common agreement of how to describe the concept of and specify formal metrics for reusability at least in Component-based Software Engineering (CbSE) [23, 25], a clear understanding of reusability in Service-oriented Software Engineering (SoSE) has not yet been reached. Although initial work exists that regards reusability as a key concept in SOSE [23, 26, 27], the definition of formal metrics that can directly be used within typical modeling languages supporting service-oriented analysis or service-oriented design (e.g., SoaML [17]) is still missing. To impair this situation, reusability becomes important considering the different viewpoints towards SOA.

To address this situation, this paper delivers initial contributions: First, we introduce refined aspects of an abstracted development process for SOSE in which we identify activities that deal with reusability and discuss characteristics of SoaML elements relating to reusability. Second, we present selected metrics measuring reusability of specific service analysis or service design models. The presented metrics are based on common agreement of how to describe and measure reusability of software artifacts on a conceptual level [4, 18, 19, 20]. Third, we demonstrate the application of these metrics in a real world scenario dealing with the construction of reusable services for a distributed management system that is based on reusable management services [2, 3, 6, 7].

The remaining parts of this paper are structured as followed: in Section 2, we outline the background of reusability in service-oriented architecture and summarize related work. Section 3 presents a typical service-oriented development process that is focused to consider aspects of designing reusable management services. In Section 4, the main contribution of this paper is introduced: we discuss three different aspects of reusability of management services in detail and present formal metrics to evaluate the respective aspect. Section 5 embeds the presented metrics in a real world development process considering management services supporting a typical Incident Management process. We chose to demonstrate the applicability of our approach within a very special scenario as future management systems will greatly benefit from applying service-orientation [10, 11, 12, 13, 14, 22, 24] thus require proper designed management services according to reusability aspects.

Finally, Section 6 summarizes the results of this paper and presents some ongoing work that can complement the proposed approach.

II. BACKGROUND AND RELATED WORK

As a special instance of distributed information systems, constructing software systems supporting IT Service Management (ITSM) follows similar principles. Considering the construction of such systems, the main challenges that distributed management is faced with are named in [10, 11, 12, 22, 24]. Although there exist a few holistic approaches considering technical aspects of distributed management based on web services [30, 31], only a few papers have been published dealing with more process-oriented aspects of integrated management systems [6, 7, 13, 14, 32]. One can conclude that, although initial work towards standardized and reusable management services has been performed, a revision of these approaches contributing to the process of Service-oriented Software Engineering (SoSE) is necessary. As we currently observe a shift towards web-based usage of dynamic IT Services ("Cloud Computing") with the broader adoption of flexible service infrastructures by the business, this holds even more. Standards such as ISO/IEC20000-1:2005 [5] only serve as a starting point.

According to [4], reusing software is the process of creating software systems from existing software rather than building software systems from scratch. Thus, from the perspective of a software designer tasked to create a collaborative system, using for instance deployed artifacts is a building block to create a system that fulfills requirements that are subject-specific. Focusing this generally applicable assumption to the challenge of creating a collaborative system supporting management activities, applying the principles of service-orientation perfectly seems to fit these requirements.

Initial work has been published lately considering design issues of reusable services [23, 26], however, investigating reusability of software artifacts is a much more older research topic and is based on concepts that were introduced at the NATO Software Engineering Conference in 1968 [35]. Many research efforts have been undertaken to address different aspects of reusability (e.g., in Component-based Software Engineering [36]), including extensive survey papers [4] that conclude the then leading insights.

Following Krueger, four different criteria have to be regarded considering reusability: *abstraction*, *selection*, *specialization* and *integration* [4]. While these concerns are of very generic nature, Erl introduces four extra criteria focused on designing service-oriented software artifacts [27]: *agnostic from business processes*, *generic business logic*, *generic service contract* and *concurrency*. However, the presented criteria in [27] are discussed on a conceptual level without any formal defined foundation. Besides many more, Poulin addresses reusability focused on object-oriented software design [19] by focusing the criteria *cohesion*, *autonomy*, *usefulness* and *complexity*. While the discussed criteria serve as a direct foundation to investigate criteria for service-oriented design, Poulin mainly focuses on business-related aspects thus considering economic measures rather than engineering measures.

Apart from the beforehand named criteria, generic aspects desired when constructing software components such as *complete* operation *sets* or *disjoint operation sets* are mainly motivated by practical concerns derived from experience we observed in several development projects. While the criterion *complete operation sets* aims at reducing future development efforts by explicitly extending service design based on predefined patterns, focusing *disjoint operation sets* tries to prevent the definition of redundant operations thus leading to side effects when changing existing service logic.

III. A SERVICE-ORIENTED DEVELOPMENT PROCESS

Designing distributed software systems is a highly complex issue that involves several different stakeholders. Focusing Service-oriented Architecture (SOA), some generic steps and models can be identified that are independent of concrete development process models. In order to utilize the metrics framework presented afterwards, in this chapter we briefly discuss an abstracted view of such typical development models. The abstracted view is presented in means of a scenario within a typical IT Service Provider (ITSP) that aims at automating its management processes based on its existing management tools. The integrative artifacts typically are implements using web services. Figure 1 shows for an overview of the assumed scenario.

As the proposed standard language for modeling service-oriented software systems already is adopted by tool vendors, the entire development process is supported by SoaML [17] for modeling services and OWL [37] for the definition of ontologies [8]. We propose to utilize OWL ontologies for defining domain models as according to [1] this approach brings several advantages focusing model-driven software development.

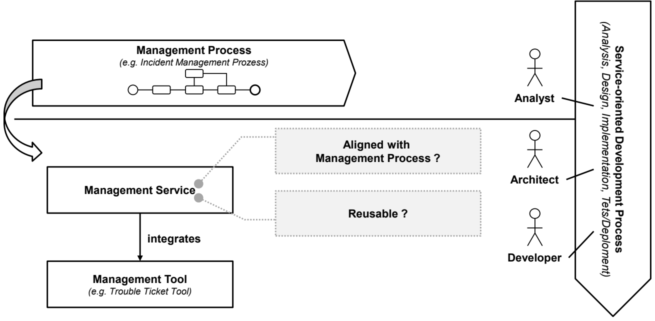

Management Process
(e.g. Incident Management Prozess)
Analyst
Aligned with
Management Process ?
Architect
Management Service
Reusable ?
integrates
Developer
Management Tool
(e.g. Trouble Ticket Tool)
Service-oriented Development Process
(Analysis, Design, Implementation, Tets/Deploment)

Figure 1. Service-oriented integration of managemen tools

The service-oriented design process discussed here is derived from and aligned with established software development processes and thus consists of the four phases *service-oriented analysis*, *service-oriented design*, *service implementation* and *deployment*.

The goal of service-oriented analysis is to capture the characteristics and requirements of the problem domain and transform them into a set of service candidates providing the necessary functionality.

To accomplish this, the analyst will first specify a domain ontology, serving not only as the basis for the following analysis steps but also as a reference point for activities throughout the entire development process, such as the evaluation of reusability conducted mainly in the design phase. The domain ontology is an extension of a common and binding domain meta model [2, 3], ensuring consistent syntax and semantics across multiple projects and development teams.

The next step is the specification of the high-level system behavior through the definition of formal business process models that refer to the concepts found in the domain model. Focused in our domain, we define management processes as special instances from generic business processes. From these management process models, service candidates will be derived according to the rules defined in [2], that, represented as SoaML Capabilities, mark the transition to the service-oriented design phase.

The rule-based transition from service candidates to abstract (e.g. platform-independent) service interfaces constitutes the first step in the development of the service interface model. It is followed by the specification of service contracts, participants and the overall architecture.

Our evaluation of the resulting services' reusability mainly takes place towards the end of the analysis phase and early in the design phase and is based on the service candidate and service interface models. This way, achieving high levels of reusability should become more likely, since the effort required to modify analysis or design models is relatively small compared with the modification of fully implemented software.

In the implementation phase, the abstract service interfaces are concretized using platform-specific interface definition languages such as the Web Services Description Language (WSDL) [28]. Basic services are realized through implementation in code or through integration of pre-existing tools; for composite services mechanisms like the Business Process Execution Language (BPEL) [29] may be used.

These activities however, as well as the subsequent deployment phase, are not covered by our research.

IV. Reusability of Management Services

This section explores three of the aforementioned criteria for the reusability of services in greater detail and tries to establish a formal basis for their evaluation in concrete scenarios. The presented criteria are based on previously published work that, although targeting Object-oriented or Component-based Software Engineering, refines these approaches by explicitly addressing characteristics of a service-oriented design process.

A. Classification

To be able to discover the services to be reused in a specific context is essential for the process of selection and thus for reusability itself. In other words: "To reuse a software artifact effectively, you must be able to 'find it' faster than you could 'build it'." [4].

Classification is the non-technical aspect of discoverability; the technical aspect being the existence of some kind of service repository (such as Universal Description, Discovery and Integration (UDDI) [33]) supporting the actual retrieval of services.

While one can locate the services needed in a given scenario based on their name (which is, in fact, greatly facilitated by adhering to naming conventions), an extensive classification allows for a more precise search. The proposed classification categorizes the developed services according to the structure of the underlying domain model and thus allows us to locate and compare services based on a variety of different characteristics. Classifications can be defined in SoaML models using so-called Categories, that are mainly an extension of the OMG-defined Reusable Asset Specification (RAS) [16]. A SoaML Category contains several different SoaML Categorization elements that can be used to define a certain aspect.

Although we evaluate the reusability of services during design time, in order to be able to locate the implemented services the classification must pertain to them as well. We therefore assume that the SoaML Categorization elements used to classify design related artifacts are transformed into an appropriate semantic annotation of the resulting concrete service interfaces, such as Semantic Annotations for WSDL (SAWSDL) [34].

The classification dimensions for service candidates (represented by SoaML Capabilities) and service interfaces (represented by SoaML ServiceInterfaces) are shown in Table 1.

TABLE I. CLASSIFICATION DIMENSIONS

Symbol	Description
MST	*Management Service Type* Type of the service (Basic or Composed)
MCT	*Management Capability Type* Type of the specified capability (Provided or Required; only applies to service candidates)
MAT	*Management Area Type* Management Area the service belongs to
ME	*Management Entity* Entity the service operates on

To gain a measure for the extent of classification for a given service candidate MSC_x we divide the amount of classification dimensions associated with the service ($\mathrm{AoCD}(MSC_x)$) by the total number of applicable classification dimensions ($\mathrm{ToCD}_{\mathrm{SC}}$). Equation (1) defines the ratio of classification dimensions, and (2), (3), (4) and (5) define several helper functions.

$$\mathrm{RoCD}_{\mathrm{SC}}(MSC_x) = \frac{\mathrm{AoCD}(MSC_x)}{\mathrm{ToCD}_{\mathrm{SC}}} \tag{1}$$

where

$$\mathrm{AoCD}(MSC_x) = \sum\nolimits_{\forall i} \mathrm{Ex}(CD_i, MSC_x) \tag{2}$$

with

$$\mathrm{Ex}(CD_i, MSC_x) = \begin{cases} 1, \text{ if } \exists c \in CD_i \text{ associated with } MSC_x \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

and

$$\mathrm{ToCD}_{\mathrm{SC}} = |\{CD_i \mid \forall i\}| \tag{4}$$

$$\mathrm{ToCD}_{\mathrm{SI}} = |\{MCT, MST, MAT, ME\}| = 4 \tag{5}$$

The extent of classification for the actual service interface is calculated accordingly. Equation (6) gives the definition of the ration of classification dimensions for an actual service interface and (7) defines the needed dimension set.

$$\mathrm{RoCD}_{\mathrm{SI}}(MSI_x) = \frac{\mathrm{AoCD}(MSI_x)}{\mathrm{ToCD}_{\mathrm{SI}}} \tag{6}$$

with

$$\mathrm{ToCD}_{\mathrm{SI}} = |\{MST, MAT, ME\}| = 3 \tag{7}$$

An RoCD value of 1 implies an optimal classification coverage for the evaluated artifact (service candidate or service interface) in that there exists an association with at least one classification element from each available classification dimension. Values closer to 0 on the other hand indicate a relatively poor classification coverage, which is not desirable.

B. Complete operation sets

Many commonly used interface operations appear in groups, such as the well-established CRUD pattern or operation pairs like *open/close*. Completeness regarding such patterns benefits reusability, because it is very likely that, once one of the operations contained in a group is needed, all of them will be at some point.

Ignoring completeness patterns can lead to uncontrolled extension of service interfaces resulting in a loss of cohesion (if additional functionality is assigned to a separate interface) or even non-disjoint functional contexts (if functionality is unintentionally duplicated).

The sub-criteria for the completeness of data-centric services — in our context called *Entity Services* [27] — are the existence of a *Create*, *Read* and *Update* method, captured by (8):

$$\mathrm{Ex}(CO, MSC_x),\ \mathrm{Ex}(RO, MSC_x),\ \mathrm{Ex}(UO, MSC_x) \tag{8}$$

Since ISO/IEC20000-1:2005 [5] — on which the motivating example is based — does not permit the deletion of records, we do not consider the existence of a *Delete* operation necessary for interface completeness.

Thus, the measure for the completeness of an Entity Service is defined in the following equations (9) and (10)

$$\mathrm{RoC}_{\mathrm{SC}}(MSC_x) = \frac{\mathrm{Ex}(CO, MSC_x) + \mathrm{Ex}(RO, MSC_x) + \mathrm{Ex}(UO, MSC_x)}{3} \tag{9}$$

$$\mathrm{RoC}_{\mathrm{SI}}(MSI_x) = \frac{\mathrm{Ex}(CO, MSI_x) + \mathrm{Ex}(RO, MSI_x) + \mathrm{Ex}(UO, MSI_x)}{3} \tag{10}$$

where a value of 1 indicates completeness of the service regarding the CRU pattern and values below 1 indicate lacking completeness.

It should be mentioned that the concept of completeness can also be applied to data types (e.g., by demanding the existence of an ID attribute), although this paper does not further investigate this.

C. Disjoint operation sets

Like other, more "traditional" software systems, service-oriented architectures depend on the separation of concerns and on clearly defined functional borders. Those concepts can have a positive effect on reusability insofar as they structure the collection of available services and help to alleviate the problems arising from duplicated functionality such as productivity losses and potential incompatibilities and access conflicts.

In this context, one policy that is both easy to define and easy to enforce is the exclusive data access of Entity Services, implying that the access to one class of entities is to be provided by the corresponding Entity Service alone.

A statement about two services being disjoint can be made by determining the overlap of their respective operations. Assuming we have a means to decide whether two operations are functionally equivalent, defined by (11)

$$\mathrm{Cov}(O_1, O_2) = \begin{cases} 1, \text{ if } O_1 \text{ and } O_2 \text{ are equivalent} \\ 0, \text{ otherwise} \end{cases} \tag{11}$$

we can derive a measure for disjoint operation sets of services as followed in (12):

$$\mathrm{AoSSO}_{\mathrm{SC}}(MSC_x) = \sum\nolimits_{i=0}^{i<|\{O_x\}|} \sum\nolimits_{j \neq x} \sum\nolimits_{k=0}^{k<|\{O_j\}|} \mathrm{Cov}\left(O_{x,i}, O_{j,k}\right) \tag{12}$$

As this returns the total number of one service's operations also found in other services, a value of 0 (indicating completely disjoint services) should be targeted.

V. Designing reusable services for Incident Management

The following example is an excerpt from a project involving the introduction of an IT-supported incident management process. It shows the refinement of one basic service through the analysis, design, and implementation phases of the development process discussed in Section 3.

Figure 2 shows the relevant parts of a domain ontology created in accordance with ISO/IEC20000-1:2005 [5]. As depicted, two management activities (*RecordIncident* and *CreateIncidentRecord*) are regarded. The two management activities belong to different types of management activities but can be ranged in the management area IncidentManagement. The basic management activity *CreateIncidentRecord* has access to the management entity *IncidentRecord*. The presented domain ontology was defined using OWL.

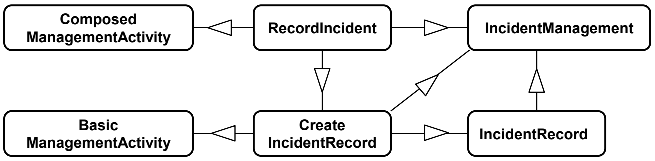


Figure 2. Excerpt from domain ontology

For the sake of simplicity, we exclusively consider the management activity for recording an incident, requiring the basic capability to create an *IncidentRecord* entity. The service operation *CreateIncidentRecord* is assigned to a SoaML Capability named *IncidentRecordService* and categorized as *RequiredManagementCapability* (denoting a needed as opposed to an already existing service) (see Figure 3).

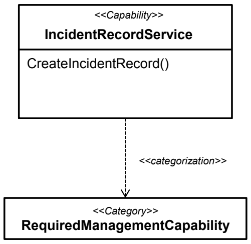


Figure 3. Preliminary service candidate

A preliminary evaluation of the service candidate's reusability shows that the criteria complete operation sets and disjoint operation sets are not fulfilled yet, as indicated by the following applications of the presented metrics:

$$\mathrm{RoCD}_{\mathrm{SC}}(IRS) = \frac{0+1+0+0}{4} = \frac{1}{4}$$

and

$$\mathrm{RoC}_{\mathrm{SC}}(IRS) = \frac{1+0+0}{3} = \frac{1}{3}$$

Consequently, the SoaML Capability is further categorized as a *ManagementBasicService* (it is, in fact, a *ManagementEntityService*, a special kind of *ManagementBasicService*), operating on *IncidentRecord* entities and belonging to the area of *IncidentManagement*. Furthermore, it is completed with respect to the CRU pattern by adding the operations *ReadIncidentRecord* and *UpdateIncidentRecord*, resulting in

$$\mathrm{RoCD}_{\mathrm{SC}}(IRS) = 1$$

and

$$\mathrm{RoC}_{\mathrm{SC}}(IRS) = 1$$

The modified service candidate can be seen in in Figure 4.

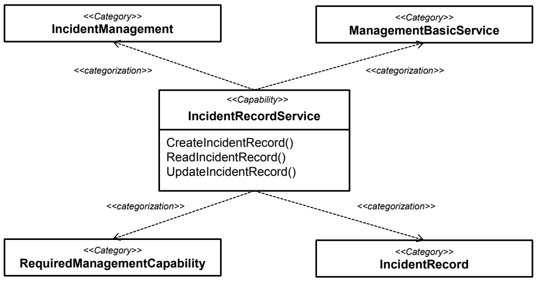


Figure 4. Modified service candidate

Since this example focuses on one single service, its operations cannot be compared to those of other services. Following the data sovereignty policy on the other hand ensures that operations managing the lifecycle of *IncidentRecords* are only found on *IncidentRecordService*. It follows that

$$\mathrm{AoSSO}_{\mathrm{SC}}(IRS) = 0$$

The ServiceInterface named *IncidentRecordService* (Figure 5) is derived from the refined Capability, whose classification it shares (with the exception of *RequiredManagementCapability*, which only applies to service candidates). Its operations are provided with appropriate parameters and return values.

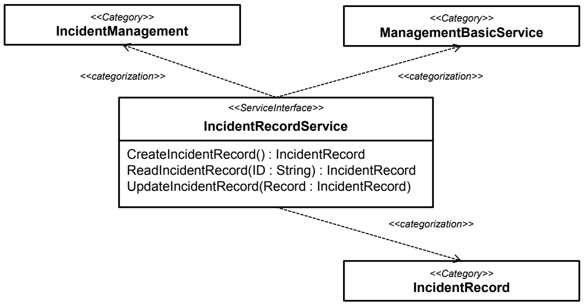


Figure 5. Service Interface

Not surprisingly, the results of the reusability evaluation are the same as for the service candidate:

$$\mathrm{RoCD}_{\mathrm{SI}}(IRS) = 1$$

$$\mathrm{RoC}_{\mathrm{SI}}(IRS) = 1$$

$$\mathrm{AoSSO}_{\mathrm{SI}}(IRS) = 0$$

The actual implementation of the *IncidentRecordService* will be achieved by the adapter-based integration of *Mantis BugTracker* [21], a trouble ticket tool currently in use as a standalone solution.

VI. Conclusion and Outlook

Reusing existing software assets seems to many researchers a kind of Holy Grail when engineering complex and distributed information systems. While a couple of different approaches and paradigms have been proposed in the past (ranging from simple Client/Server computing to up to Component-based Software Engineering (CbSE)) to address general problems when reusing software assets, many issues still remain. Grounded on the simple statement, that reusing does not come for free [19], special attention has to be paid not only within the design process but also when selecting appropriate models and modeling techniques.

As nowadays information systems have to be aligned with business processes, the requirements for reengineering business logic can directly be derived from the business process perspective. Considering Service-oriented Architectures (SOA) to realize these process-oriented information systems, the systems elements that implement SOA have to be aligned with the business processes. Thus, reusability of services has to be regarded from the perspective of the technical-independent processes. Existing approaches do not consider process requirements explicitly when targeting the design of reusable services but mainly focus on technical details.

To address this issue, in this paper we deliver several contributions. First, we refine a generic development process for service-oriented analysis and design and outline development tasks that are supported by different models and modeling techniques that focus the reusability of the to-be-designed artifacts. The presented development approach extends and refines work that was previously published by our research group [2, 3]. Second, we discuss an assorted selection of different aspects of reusability considering service orientation and present three different aspects that are formalized using a conceptual metrics framework. The presented metrics can be applied to any kind of service analysis or design models if they are defined using SoaML. Furthermore, we outline the advantage of using an OWL-based domain ontology for directly influencing the quality of service design. Using domain ontology has several advantages [2, 3]. As we expect that reusability of services can only be discussed within clearly defined domains, we present an application example of our approach within the context of designing web service-based services for a process-oriented management system supporting IT Service Management Processes. A third contribution therefore devotes to a typical Incident Management process and demonstrates the application of both the presented development process and the introduced metrics framework, resulting in a set of management services that are designed along special design characteristics.

Although service-oriented computing inherently is predestinated for building software systems based on existing assets, it seems remarkable that existing approaches mainly focused on technical details. Considering the contributions we deliver in this paper, we address a more conceptual perspective, but further work has to be performed. As we mainly focused on generic issues targeting design related aspects of services reusability, a more formal approach that is independent of certain domains could greatly enhance software engineering. Utilizing model-driven techniques could not only decrease engineering round trip times, but also increase the quality of resulting systems implementation.

References

[1] D. Gasevic, D. Djuric, and V. Devedzic, Model Driven Engineering and Ontology Development, Springer, Heidelberg, 2006.

[2] I. Pansa, F. Palmen, S. Abeck, K. Scheibenberber, “A Domain-driven Approach for Designing Management Services”, SERVICECOMPUTATION2010, Lisbon, 2010.

[3] I. Pansa, P. Walter, K. Scheibenberger, and S. Abeck, “Model-based Integration of Tools Supporting Automatable ITSM Processes”, IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), 2010, Page(s): 99 – 102.

[4] C.W. Krueger, “Software reuse”, *ACM Computing Surveys (CSUR)*, vol. 24, no. 2, pp. 131-183, 1992.

[5] ISO/IEC20000-1:2005, Information technology – Service management – Part1: Specification, International Standards Organization (ISO), 2005.

[6] G. Aschemann and P. Hasselmeyer, „A Loosely Coupled Federation of Distributed Management Services“, Journal of Network and Systems Management, Vol 9, No. 1, 2001.

[7] N. Anerousis, „An Architecture for Building Scalable, Web-Based Management Services“, Journal of Network and System Management, Vol. 7, No 1., 1999.

[8] T. Gruber, „A translation approach to portable ontology specifications“, Knowledge Acquisition, Vol. 5, Issue 2, pp. 199-220, 1993.

[9] M. Henning, „The Rise and Fall of CORBA“, Communications of the ACM Vol. 51 No. 8, pp. 52-57, 2008.

[10] P. Kumar, „Web Services and IT Management“, ACM Queue, Volume 3 Issue 6, 2005.

[11] V. Machiraju, C. Bartolini and F. Casati, „Technologies for Business-Driven IT Management“, in Extending Web Services Technologies: the Use of Multi- Agent Approaches, Kluwer Academic, 2004.

[12] A. Moura, J. Sauve and C. Bartolini, „Research Challenges of Business-Driven IT Management“, 2nd IEEE/IFIP International Workshop on Business- Driven IT Management, pp.19-28, 2007.

[13] C. Mayerl, F. Tröscher, and S. Abeck, „Process-oriented Integration of Applications for a Service-oriented IT Management“, The First

International Workshop on Business-Driven IT-Management, BDIM'06, pp. 29-36, 2006.

[14] C. Mayerl, T. Vogel and S. Abeck, „SOA-based Integration of IT Service Management Applications“, 2005 IEEE International Conference on Web Services (ICWS 2005), 2005.

[15] Organization for the Advancement of Structured Information Standards (OASIS), Reference Model for Service Oriented Architecture, Version 1.0, OASIS, August 2006.

[16] Object Management Group (RAS), Reuasable Asset Specification, Version 2.2, OMG, November 2005.

[17] Object Management Group (OMG), Service-oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS), FTF Beta 1, OMG, April 2009.

[18] R. Prieto-Diaz and P. Freemann, „Classifying Software for Reusability“, IEEE Software, Vol. 4, Issue 1, pp. 6-16, 1987.

[19] J. Poulin, „Measuring Software Reuse“, Addison Wesley Publishing Group, 1997.

[20] Ruben Prieto-Diaz, „Status Report Software Reusability“, IEEE Software, Vol. 10, Issue 3, pp. 61-66, 1993.

[21] Mantis Bug Tracker, 2011; http://www.mantisbt.org/ (last vistited: 19- 02-2011).

[22] J. Sauvé, A. Moura, M. Sampaio, J. Jornada and E. Radziuk, „An Introductory Overview and Survey of Business-Driven IT Management“, 1st IEEE/IFIP International Workshop on Business-driven IT Management (BDIM 2006), 2006.

[23] A. Sillitti, and G Succi, „Reuse: From Components to Services“, High Confidence Software Reuse in Large Systems (2008), pp. 266-269, 2008.

[24] V. Tosic, „The 5 C Challenges of Business-Driven IT Management and the 5 A Approaches to Addressing Them“, The First IEEE/IFIP International Workshop on Business-Driven IT Management, 2006.

[25] H. Washizaki, H. Yamamoto, and Y. Fukazawa, „A Metrics Suite for Measuring Reusability of Software Components“, Proceedings of the 9th International Symposium on Software Metrics, 2003.

[26] J. Wang, J. Yu, P. Falcarin, Y. Han, and M. Morisio, „An Approach to Domain- Specific Reuse in Service-Oriented Environments“, Proceedings of the 10th international conference on Software Reuse: High Confidence Software Reuse in Large Systems, 2008.

[27] T. Erl: SOA, Principles of Service Design, Prentice Hall, 2008.

[28] World Wide Web Consortium (W3C), Web Service Description Language (WSDL) Version 2.0 Part1 Core Language, W3C Recommendation, 2007.

[29] Organization for the Advancement of Structured Information Standards (OASIS), Web Services Business Process Execution Language (WS- BPEL), Version 2.0, OASIS, April 2007.

[30] Organization for the Advancement of Structured Information Standards (OASIS), Web Services Distributed Management: MUWS Primer, OASIS, Februar 2006.

[31] Distributed Management Task Force (DMTF), Web Services for Management (WS-Management) Specification, Version 1.1.0, DMTF, 2010.

[32] G. Tamm and R. Zarnekow, „Umsetzung eines ITIL-konformen IT-Service-Support auf der Grundlage von Web-Services“, Wirtschaftsinformtik 2005, pp. 647-666, 2005.

[33] Organization for the Advancement of Structured Information Standards (OASIS): Universal Description, Discovery and Integration (UDDI), Version 3.0.2, Oktober 2004.

[34] World Wide Web Consortium (W3C): Semantic Annotations for WSDL and XML Schema, W3C Recommendation, 2007.

[35] P. Naur and B. Randall, “Software Engineering; Report on a conference sponsored by the NATO Science Committee”, Garmisch, 1968.

[36] B. Meyer, “Reusability: The case for Object-oriented design”, in Frontier Series: Software Reusability, Vol. 11 – Applications and Experience, 1989, pp. 1-33.

[37] World Wide Web Consortium (W3C), Web Ontology Language (OWL), W3C Recommendation, 2004.

ISBN: 978-1-61208-152-6

Evaluating Service-oriented Vendor Platforms with a Dedicated Architecture Maturity Framework

Helge Buckow
McKinsey & Company
SOA Innovation Lab
Berlin, Germany
helge_buckow@mckinsey.com

Hans-Jürgen Groß
Daimler AG
SOA Innovation Lab
Stuttgart, Germany
hans-juergen.gross@daimler.com

Oliver F. Nandico
Capgemini
SOA Innovation Lab
Munich, Germany
oliver.f.nandico@capgemini.com

Gunther Piller
University of Appl. Sciences Mainz
SOA Innovation Lab
Mainz, Germany
gunther.piller@fh-mainz.de

Karl Prott
Capgemini
SOA Innovation Lab
Hamburg, Germany
karl.prott@capgemini.com

Alfred Zimmermann
Reutlingen University
SOA Innovation Lab
Reutlingen, Germany
alfred.zimmermann@reutlingen-university.de

***Abstract* – The SOA Innovation Lab - an innovation network of industry leaders in Germany and Europe - investigates the practical use of vendor platforms in a service-oriented context. As a part of this investigation the SOA capabilities of products from different vendors need to be evaluated. For this purpose a service-oriented architecture evaluation framework has been developed and currently extended, leveraging and extending CMMI and TOGAF, as well as other service-oriented state-of-the art frameworks and methods. Besides details about our evaluation framework, we present and analyze results of various service-oriented platforms from assessments with four major vendors. Our idea and contribution is to extend existing Service Oriented Architecture (SOA) maturity frameworks to accord with a sound metamodel approach. Our metamodel for architecture evaluation is based on the well understood and standardized Capability Maturity Model Integration (CMMI), which was originally used to assess software processes and not architectures. Our specific architecture capability evaluation approach is the result of a metamodel-based analysis and synthesis from state of art models. The paper presents an original approach for systematically and cyclic evaluations of heterogeneous service-oriented platforms in practical use.**

Keywords – Evaluation; SOA Vendor Platforms; SOA Maturity Model; SOAMMI; CMMI; TOGAF; Assessment Questionnaire; Framework Validation; Results; Key Findings.

I. INTRODUCTION

The growing complexity of IT landscapes is a challenge for many companies. A large number of packaged solutions platforms - mostly extended and modified - individual software solutions, legacy applications, and different infrastructure components lead to high cost and limited ability to respond quickly to new business requirements. Many companies start enterprise architecture management [1] and [2] (EAM) initiatives to address this problem. In areas where flexibility or agility in business are important, SOA is the approach of choice to organize and utilize distributed capabilities. Here, the use of standard software [3] is often a challenge, in particular when dealing with services on a fine granular level.

Initially SOA was burdened with hype and inflated expectations. Now it is part of an ongoing discussion about software architecture. The benefits of SOA are recognized. They comprise flexibility, process orientation, time-to-market, and innovation. The adoption of tools and methods for SOA is growing. An overview about the current status of SOA adoption and reports on the maturity of SOA technology from vendors is provided by [4], [5], and [6].

To analyze the SOA ability of major vendor platforms in a systematic way, the SOA Innovation Lab has developed a questionnaire-based assessment method based on a specifically designed SOA architecture maturity framework to support the fundamental evaluation method [7]. The latter was constructed by integrating different analysis approaches for architecture dimensions, using a consistent meta-model based on correlation analysis of intrinsic model elements. Details about this framework, the corresponding questionnaire and general findings from consecutive assessments with four major vendors are focus of this paper.

Our SOA architecture maturity framework is part of an approach for the design of a service-oriented enterprise architecture with custom and standard software packages, which we briefly sketch in the following (for details see [3]): The method starts with the definition of company domain maps, identifying in particular areas where SOA benefits - like agility, flexibility, and reduction of redundancies - are a priority. As a next step one needs to define and decompose the services for these domains, to identify standardisable services. On this basis one is able to decide whether standard platforms should be used within a SOA architecture.

In addition to the overall decision, whether a standard platform should be used within a certain domain, it is necessary to map a vendor solution to a company's domain map and its services, to evaluate the overall functional fit.

ISBN: 978-1-61208-152-6

For this, the SOA ability of the identified package needs to be evaluated against specific SOA use cases.

For software services that fulfill the functional and non-functional requirements, a target-architecture needs to be developed. It includes the high-level system architecture, as well as integration patterns for the physical integration of systems (see also [8] and [4]). The SOA Innovation Lab has developed a capability map for integration that allows, to structure corresponding requirements. In addition, a taxonomy for integration patterns and corresponding best practices has been compiled. Finally, a business-case-oriented implementation was defined and is currently under development and evaluation.

In this paper, we provide in Section II details about the related work background models leading to our SOA architecture maturity framework in Section III. Section IV summarizes the derivation of an assessment questionnaire for vendor workshops. Initial results from the evaluation of four major SOA vendor platforms are presented in Section V. In Section VI, we draw conclusions and sketch future developments.

II. Background and Model Integration

Enterprises need to systematically evaluate opportunities from a potential investment in SOA with standard platforms in heterogeneous IT-environments. For this purpose we have combined a consistent metamodel for assessing transcend disciplines, with content elements from holistic enterprise architecture frameworks, which comprises all important architecture dimensions.

Regarding the metamodel we have built upon CMMI [9], which is originally an assessment framework for software processes and not for enterprise software architectures. To transform CMMI into a specific framework for the assessment of the maturity of enterprise and software architectures, we have originally combined CMMI with current architecture framework and maturity models. Our approach is more generally and different to ATAM [10], which is an architecture evaluation process, based on risk-adapted definable quality goals and fine granular architecture requirements. In particular we use TOGAF [1] as a basic structure for enterprise architecture, spanning all relevant enterprise and software architecture types.

Of course, TOGAF is missing as a general standard important architecture detail structure and doesn’t cover all investigated architecture domains. In addition, we have cross checked and – if appropriate - extended our model with supporting elements from the following state of art SOA maturity models, and with our original model integration extensions, which are mentioned in Section III.

The Architecture Capability Maturity Model (ACMM) [11] framework, which is included in TOGAF, was originally developed by the US Department of Commerce. The main scope of ACMM is the evaluation of enterprise architectures in internal enterprise architecture assessments. The goal of ACMM assessments is to enhance enterprise architectures by identifying quantitatively weak areas and to follow an improvement path for the identified gaps of the assessed architecture. The ACMM framework consists of six maturity levels and nine specific architecture elements ranked for each maturity level - deviant from CMMI. SOAMMI was influenced by some definitions of ACMM for basic maturity levels of enterprise architecture.

The SOA Maturity Model of Inaganti/Aravamudan [12] considers the following multidimensional aspects of a SOA: scope of SOA adoption, SOA maturity level to express architecture capabilities, SOA expansion stages, SOA return on investment, and SOA cost effectiveness and feasibility. The scope of SOA adoption in an enterprise is differentiated by following levels: intra department or ad hoc adoption, interdepartmental adoption on business unit level, cross business unit adoption, and the enterprise level, including the SOA adoption within the entire supply chain. The SOA maturity levels are defined related but different to CMMI using five ascending levels to add enhanced architectural capabilities: level 1 for initial services, level 2 for architected services, level 3 for business services, level 4 for measured business services, and level 5 for optimized business services. In a two-dimensional view - SOA scope and SOA maturity level - proper expansion stages for the systematic introduction of SOA in an enterprise are differentiated: fundamental SOA in a local department view, networked SOA with architected services on business unit level, and process enabled SOA on the enterprise level or in conjunction with suppliers.

The SOA Maturity Model from Sonic [13] distinguishes five maturity levels of a SOA, and associates them in analogy to a simplified metamodel of CMMI with key goals and key practices. Key goals and key practices are the reference points in the SOA maturity assessment. We mention the following Key Goals: institutionalize use of SOA, put in place architecture leadership for SOA, and prove returns from use of standard technologies, which have influenced the definition of the Maturity Level 2 (Managed) of SOAMMI.

The SOA Maturity Model of ORACLE [14] characterizes in a loose correlation with CMMI five different maturity levels: opportunistic, systematic, enterprise, measured, industrialized and associates them with strategic goals and tactical plans for implementing SOA. Additionally following capabilities of a SOA are referenced with each maturity level: Infrastructure, Architecture, Information & Analytics, Operations, Project Execution, Finance & Portfolios, People & Organization, and Governance. The Maturity Level 2 (Systematic) of the SOA Maturity Model from ORACLE has influenced technical views on Architecture Areas within the Application Architecture and the Technology Architecture Domain of SOAMMI specifying important SOA infrastructures like initial project level use of ESB and BPEL for service integration and orchestration, service-level access to information sources, enterprise applications through standards for Web Services: WSIF, JCA, JMS, initial use of service registry, basic service management infrastructure for monitoring and declarative application of runtime policies, e.g., message level security.

III. SOAMMI - ARCHITECTURE MATURITY FRAMEWORK

To enable corresponding assessments of the SOA ability of standard software, we have originally extended our SOA architecture maturity framework - *SOA Maturity Model Integration* (SOAMMI) from [7] and added architecture classification models and architecture evaluation and integration patterns. In respect to requirements from customer oriented domain models and reference use scenarios, our SOAMMI architecture maturity framework introduces the following originally defined maturity levels, which define important quality criteria for software and enterprise architecture excellence and help to measure the architecture maturity of vendor products:

1. *Maturity Level: Initial*

The Initial Level is the entry level of architecture maturity. Here the vendor service architecture is incomplete or with no or initial coverage related to the customer demand. The architecture is unpredictable and poorly controlled. The software architectures are ad hoc and chaotic. The assessed software organization does not provide a stable environment to support software and enterprise architectures.

2. *Maturity Level: Managed*

Projects of managed organizations have ensured that architectures are planned and executed in accordance with an architecture policy. Projects typically employ skilled architects who have adequate resources to produce controlled outputs. Software architectures are monitored, controlled, reviewed and evaluated from time to time, for adherence with architecture standards.

3. *Maturity Level: Defined*

Architectures are well characterized and understood, and are rigorously described in standards, procedures, tools, and methods. The service architecture of the software technology vendor is defined, having large, increasing completeness and coverage. An organization's set of architecture standards is established and improved over time. The customer service architecture is agile tailored from standard vendor architecture.

4. *Maturity Level: Quantitatively Managed*

This high mature level software organization establishes and uses quantitative objectives and architecture specific metrics / key architecture indicators for software architecture quality and architecture management performance as criteria in managing architectures. Quantitative objectives are based on the needs of the customer, end users, organization, and architecture implementers. Architecture artifacts and benefits are measured at vendor and customer side.

5. *Maturity Level: Optimizing*

The highest level maturity organization continually improves its software and enterprise architectures based on a quantitative understanding of the common causes of variation inherent in architectures. Their organizational focus is on continually improving architecture performance through incremental architecture development, innovative architecture management and technological improvements.

The top level structure of SOAMMI is organized considering five Architecture Domains adapted from TOGAF [1]: Architecture Strategy and Management, Business Architecture, Information Architecture, Application Architecture, Technology Architecture, Service & Operation Architecture, Architecture Realization.

Architecture Areas where originally derived primarily from TOGAF [1], Quasar Enterprise [5] and Essential [2], as well as from business requirements and pilot use cases defined by members of the SOA Innovation Lab. Architecture areas are the correspondent architecture structures for process areas from CMMI. We have defined 22 genuine architecture areas of SOAMMI fitting our architecture evaluation scope, but different from CMMI (see [9]) - and structured them according to standard architecture maturity levels.

SOAMMI supports both the staged representation and the continuous representations (Figure 1). The same staging rules as in CMMI apply to SOAMMI and should therefore enable the flexible adoption of both model representations: continuous - for assessing single architecture areas and staged - for assessing the whole enterprise architecture.

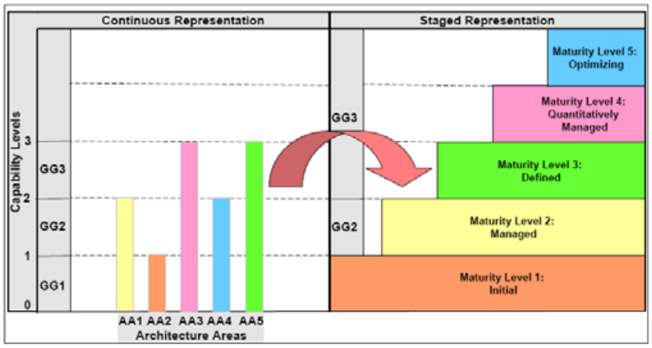

Continuous Representation
Staged Representation
Capability Levels
0
1
2
3
GG1
GG2
GG3
AA1 AA2 AA3 AA4 AA5
Architecture Areas
GG2
GG3
Maturity Level 1: Initial
Maturity Level 2: Managed
Maturity Level 3: Defined
Maturity Level 4: Quantitatively Managed
Maturity Level 5: Optimizing

Figure 1. Architecture Capability and Maturity Levels

The continuous representation of SOAMMI is similar to CMMI, which uses levels to denote the capability and the incremental improvement path for specific architecture areas. The assessment of capability levels could be applied to iterate specific architecture areas or to assess or improve a focused innovation aspect, involving one ore more architecture areas. To verify and support the persistent institutionalization of architecture areas we have introduced in the SOAMMI framework generic goals and practices.

Specific Goals describe the objectives within a single architecture area. Necessary activities associated with a specific goal are expressed through Specific Practices. As an example, within the architecture domain *Business Architecture* and the architecture area *Business Capabilities and Services* we find Specific Goals (SG) and Specific Practices (SP) like:

SG 1: Determine business services for SOA packaged software solutions and optimize business processes

- SP 1.1: Identify and map business services to business capabilities
- SP 1.2: Determine degree of coupling between services

SG 2: Analyze coverage, adaptability and functional completeness of business capabilities and services

- SP 2.1: Assess coverage of supported business services from customer perspectives
- SP 2.2: Assess adaptability and functional completeness of business services.

IV. Assessment Questionnaire Model

Vendor assessments need to address the key challenges for companies during the built-up and management of service-oriented architectures with standard software in heterogeneous IT environments. At this stage we therefore do not consider all dimensions of SOAMMI that fulfills all academic requirements, but restrict ourselves to a pragmatic approach, which can be completed in a 3-4 hour workshop with vendor experts. In the following, we describe the artifacts, which we developed for an effective vendor assessment. Then we sketch the procedure we followed in corresponding vendor workshops.

Assessments of the SOA ability of standard software packages can be viewed as an ideal mean to engage with vendors on all relevant challenges of SOA for standard software. Therefore, we did not design our assessment in form of a survey that could be filled out remotely, but rather focused on a discussion format where answers should include artifacts, cases, best practices, etc. As most questions have different relevance and meaning for different companies, our assessment is not intended to serve as a vendor ranking of any kind.

These goals imply that a pragmatic simplification of SOAMMI is required, that needs to be enriched with specific user requirements from companies using SOA in heterogeneous environments with standard platforms.

The complete SOAMMI model includes 22 architecture areas with 38 specific goals and over 122 specific practices. Answering all these questions would yield a complete picture, but it would lack the pragmatic use cases and would require more than 10 hours to complete. Still the structure is relevant, as it ensures the coverage of all important architecture areas and helps to stay focused.

To ensure practical relevance, members of the SOA Innovation Lab have collected their most important use cases for business contexts where they think SOA and standard platforms brings benefit, but where significant implementation challenges are expected. These use cases go down to the level of singular services, tools and technologies. This approach also helps to avoid generic responses from vendors on assessment questions.

Following these ideas, the basic structure of our questionnaire was taken from SOAMMI architecture areas [7] with one or more question per specific goal. Additionally we have considered and adapted from [6] SOA design questions that affect quality attributes of vendor platforms. User requirements have been consolidated and mapped against specific goals. Wherever no user requirements could be mapped, specific practices have been used to generate questions on the level of specific goals. Through this procedure each specific goal could be related to at least one concrete question.

To avoid subjective judgment, the answer to each question was ranked, using one of three distinctive levels only:

- Not fulfilled (value zero): There is no evidence or example available
- Partially fulfilled (value one): The topic of the question is addressed, but there are still apparent gaps
- Completely fulfilled (value two): the topic of the question is fulfilled as best practice.

For reasons of applicability, we simplified the SOAMMI model for the use in an assessment. First, we did not formulate questions for generic goals but concentrated on specific goals and corresponding questions. Second, the pure CMMI logic requires that a level can only be reached if all goals are completely achieved. If there is just one specific goal that could not be achieved, the corresponding maturity level cannot be reached at all. Given these constraints, most vendors would be at level 1. In order to highlight areas for improvement, we added a degree of fulfillment for maturity levels. For each maturity level, all assessment values from the questions of this level are added together. The fulfillment of a level is then indicated as the percentage of the maximal possible value for this level (i.e., number of questions per level multiplied by value two). As a result, each of the five maturity levels has a percentage of fulfillments.

Developing an assessment framework on this basis resulted in a questionnaire, which was the foundation of the assessment process with the selected vendors. Here are examples of level 2 questions with their mapping to SOAMMI, taken from the assessment questionnaire:

1. *Architecture Domain: Architecture Strategy & Management*
   Architecture Area: Requirements Management
   - What is the vendor’s internal process and governance to get manage SOA related requirements from customers and industry specific organizations?
   - How is the ideal business capability map created/generated?
   - How are requirements found/set/derived?
   - Which/what information is communicated back to the user and when?
2. *Architecture Domain: Business Architecture*
   Architecture Area: Business Domains and Capabilities
   - Where are specific SOA capabilities in the vendor capability map?
   - What SOA capabilities are requested/planned/realized?
   - Are methods available to map a vendor specific capability map to customer specific domain/capability maps?

The assessment process takes about 3 months to complete for each standard software provider overall. The first step is a Pre-Workshop (2-3 hours), in which the SOA Innovation Lab presents the background and questionnaire

ISBN: 978-1-61208-152-6

and the vendor has the opportunity to present his SOA strategy. This workshop is essential to make sure, that the vendor can identify the appropriate experts for the assessment workshop itself. Then the actual Assessment Workshop (4-6 hours) is held a few weeks later, so that the vendor has enough time to identify the experts that should participate and prepare answers. The SOA Innovation Lab then prepares the summary of the findings and presents these back to the vendor (1-2 hours). Finally, a series of follow up workshop for specific questions (3-4 hours each) is arranged with the vendor.

V. Analysis and Synthesis Results

Our experience with assessment workshops with vendors has been very positive. Each vendor showed strong interest and was happy to hear additional user views on the topic. Figure 2 shows a summary result.

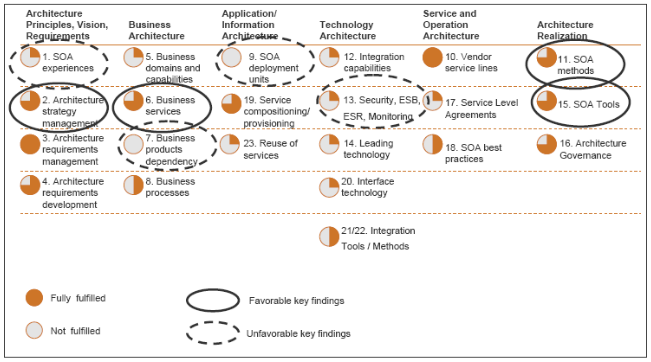

Architecture Principles, Vision, Requirements
Business Architecture
Application/ Information Architecture
Technology Architecture
Service and Operation Architecture
Architecture Realization
1. SOA experiences
5. Business domains and capabilities
9. SOA deployment units
12. Integration capabilities
10. Vendor service lines
11. SOA methods
2. Architecture strategy management
6. Business services
19. Service compositioning/ provisioning
13. Security, ESB, ESR, Monitoring
17. Service Level Agreements
15. SOA Tools
3. Architecture requirements management
7. Business products dependency
23. Reuse of services
14. Leading technology
18. SOA best practices
16. Architecture Governance
4. Architecture requirements development
8. Business processes
20. Interface technology
21/22. Integration Tools / Methods
Fully fulfilled
Not fulfilled
Favorable key findings
Unfavorable key findings

Figure 2: Overview of Vendor Workshop Results

In addition to the answers to all specific questions, we have synthesized key findings that highlight our view on the actual SOA ability of a standard platform across vendors:

SOA experiences: Even though SOA has been a topic for vendors for years now, there are no major SOA implementations that include standard software systems. Most cases have the quality of a proof of concept, often focusing on GUI integration, instead of deep functional integration. There seems to be a gap between those SOA capabilities that are offered and those, which can be actually used in a SOA.

Architecture strategy management: SOA is seen as an important part of overall strategy with no alternative in the long term. All vendors have developed SOA strategies and have integrated it into their product roadmap. In most cases, SOA enablement is a mandatory requirement for the development of new functionality.

Business Services: Vendors offer solution maps that describe the functionality in terms of services and have developed methods to find existing services to a given requirement. In addition, vendors are developing solution scenarios, which offer not just the individual service but a complete set of processes that implement a business solution.

Business product dependencies: Vendors have invested substantially in SOA, but in many cases, SOA has been only applied as wrapping of existing systems, without changing the core of the application. This means that business services are tightly coupled and therefore inflexible. Often dependencies between services were complex and could be ambiguous for the service composition.

SOA deployment units: No vendor offers licenses that allow the usage of individual services instead of the whole system. This means that users still have to purchase the whole application, which hinders a best of breed approach for composite applications.

SOA methods: There is a rich offering for methods for governance, implementation guidelines, etc. for SOA available. SOA is not just seen as the technical implementation, but rather as an engineering discipline that goes beyond service interfaces.

Security, ESB, ESR, service monitoring: Industry standards are implemented within the standard software, but standards like SAML leave room for interpretation. This makes it difficult to integrate solutions across several standard platforms, which is a requirement for most users.

SOA tools: All standard platform providers have added tool suites to their portfolio that support SOA development. The integration of these tools within development layers and across platforms is still not completely solved.

In summary, there are still obstacles to apply standard software in a heterogeneous SOA environment. Often, a vendor's SOA approach is specific to the vendor. E.g., each vendor has structured business functionality - a business domain map – defined and described in an individual way. However these business domain maps are vendor specific and often do not correlate with company specific domain maps. Vendors also often use specific semantics and data models and have incompatible technologies (ESB, repository) that do not integrate seamlessly into overall heterogeneous landscapes.

For most vendors, products are only SOA *enabled*. This means that SOA is implemented as wrapper around existing interfaces, and the internal structure is still monolithic. This typically results in a very granular and technical view (e.g., over 3.000 services) that is difficult for the user to identify and comprehend, and therefore to implement. In addition, there are many dependencies between services that often require certain modules to be implemented and populated with data, before services from other domains can be used.

Finally, most vendors have not adopted a business model that supports the usage of standard software through services. The deployment unit still is the entire software package. Individual services cannot be licensed and license models have not been adapted to service usage. Especially in a heterogeneous environment SOA service level agreements will be important, but are not established yet.

Many vendors have invested early in SOA, long before users were ready to use new SOA enabled components in an appropriate way. The investment therefore was mostly to SOA enable products from a technical point of view, without considering the business scenarios that they should support. Therefore the adoption on the user side is slow, with only a few (100-300 per vendor) pilot SOA cases, mostly focusing on GUI integration. This is a tiny fraction of the overall installed base for standard software.

The most important result from the assessment workshops with vendors is that there is a strong interest from vendors to work together with the SOA Innovation Lab to further refine SOA methods and to develop solutions for the SOA use cases. We think that this is a great asset and we will continue to build on these relationships to further develop the maturity of SOA and standard software.

The experiences from the SOA Innovation Lab show that most companies see SOA as an important part of their architecture management strategy. In order to implement this strategy in an environment with standard software from different vendors, there are key requirements that should be developed together with the vendors:

- Users need services that are as independent of their context as possible and that do not require the full implementation of the standard software, this should also be reflected in the license model.
- Individual process building blocks, that can be orchestrated to an overall business solution are key, technical interfaces come second.
- Focus on service enablement should affect areas that have high requirements for process agility and are value added for the business.

VI. Conclusion

A new method for evaluating the SOA maturity of standard software packages and its vendors has been introduced. Based on the work of CMMI - an assessment and improvement model for software processes - we have transformed and developed a suitable model for the evaluation of SOA capability and maturity. Our architecture evaluation approach was founded on the current TOGAF standard for enterprise architectures. SOAMMI – the SOA Maturity Model Integration – is the result of a metamodel-based conception and synthesis to provide a sound base for practical evaluations of service-oriented standard platforms in heterogeneous environments.

The SOAMMI framework was applied in several assessment workshops with vendors of service-oriented platforms and has provided transparent results for subsequent changes on service-oriented product architectures and related processes. It should be noted however, that the results of these assessments need to be interpreted in the context of company specific strategies and use cases. As a consequence they cannot provide vendor rankings of any kind.

Going forward, the SOA Innovation Lab plans to use SOAMMI as an ongoing framework for the cyclical evaluation of standard software packages. Based on real-world use cases and ongoing investigations in architecture evaluation patterns the framework will be continuously optimized.

Acknowledgment

The authors thanks the SOA Innovation Lab and his member companies for the joint research and practical validation of the architecture evaluation framework SOAMMI with major service technology vendors IBM, SAP, ORACLE, and Microsoft, supplemented by associated software and consulting firms and the network of scientific consultants.

References

[1] TOGAF "*The Open Group Architecture Framework*", Version 9, The Open Group, 2009.

[2] *Essential Architecture Project*, http://www.enterprise-architecture.org, last access: June, 19th, 2011.

[3] H. Buckow, H.-J. Groß, G. Piller, K. Prott, J. Willkomm, and A. Zimmermann, "*Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes*", Proc. 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010), Paderborn, 2010, GI-Edition - Lecture Notes in Informatics (LNI), P-160, 2010, pp. 219-230.

[4] T. Erl, "*SOA Design Patterns*", Prentice Hall. 2009.

[5] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.P. Richter, M. Voß, and J. Willkomm, „*Quasar Enterprise*" dpunkt.verlag, 2008.

[6] P. Bianco, R. Kotermanski, and O. Merson, "*Evaluating a Service-Oriented Architecture*", CMU/SEI-2007-TR-015, Carnegie Mellon University, Software Engineering Institute, 2007.

[7] A. Zimmermann, "*Method for Maturity Diagnostics of Enterprise and Software Architectures*", A. Erkollar (Ed.) ENTERPRISE & BUSINESS MANAGEMENT, A Handbook for Educators, Consulters and Practitioners, Volume 2, Tectum 2010, ISBN 978-3-8288-2306-8, 2010, pp. 129-172.

[8] G. Hohpe and B. Woolf, "*Enterprise Integration Patterns*", Addison Wesley, 2004.

[9] CMMI-DEV-1.3 2010 "*CMMI for Development, Version 1.3*" Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2010-TR-033, 2010.

[10] R Kazman, M. Klein, and P. Clements "*ATAM: Method for Architecture Evaluation*", Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2000-TR-004, 2000.

[11] ACMM, "*Architecture Capability Maturity Model*", in TOGAF Version 9, The Open Group Architecture Framework, The Open Group, 2009, pp. 685-688.

[12] S. Inaganti and S. Aravamudan, "*SOA Maturity Model*", BP Trends, April 2007, 2007, pp. 1-23.

[13] Sonic: "*A new Service-oriented Architecture (SOA) Maturity Model*", http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf, last access: June, 19th, 2011.

[14] Oracle: "*SOA Maturity Model*", http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet, last access: June, 19th, 2011.

Service-Oriented Architecture Concept for Intelligence Information System Development

Jugoslav Achkoski [1], Vladimir Trajkovik [2], Danco Davcev [3]
[1] Military Academy „General Mihailo Apostolski“,
Skopje, Macedonia
[2, 3] Faculty of Electrical Engineering and Information Technologies,
Skopje, Macedonia
jugoslav_ackoski@yahoo.com, trvlado@feit.ukim.edu.mk, etfdav@feit.ukim.edu.mk

***Abstract*—This paper presents an idea for Service Oriented Architecture (SOA) approach in prototype of Intelligence Information System (IIS). IIS prototype, based on SOA, could offer better coordination among institutions involved in intelligence thus providing increase of intelligence effectiveness. This approach can serve as a foundation for the establishment of the Integrated Intelligence System, which is based on services as software components. In this paper, we propose five postulates that can serve as checklist for integration of SOA in IIS.**

Keywords- concept; SOA; intelligence information systems.

I. Introduction

Intelligence, as a public service, has a great significance for a country [1]. Frequently used information systems, which support intelligence activities, have high influence in the decision making process. Modern information technology considerably contributes to the processes' (activities) improvement by supporting intelligence cycles (planning, collecting data, analyzing data and dissemination). Although, there is constant improvement in the field of information technology, significant advancement in the quality of work in the field of intelligence has not taken place in the last ten years [2].

SOA offers possibilities of making new opportunities for increasing efficiency of IIS. These opportunities could be found in a form of expanded solutions for designing intelligence and information systems [3], [4], and [5]. SOA approach in information systems is a logical solution, not only for temporary and short term exploitation, but also as a perspective solution for general strategy in companies and governmental institutions [6].

Every modern intelligence system is based on some type of information system [7]. Usage of contemporary technology, especially Information Communication Technology (ICT), is giving more efficient execution of all phases of the intelligence service.

This article is divided in several sections. Section III describes the model of Intelligence Information System. Section IV demonstrates functions of the distributed Intelligence Information System. Finally, in the Section V, expected achievements enabled by deploying SOA concept for developing intelligence information system are explained.

II. Related Work

There are numerous IT projects which dedicated their work on designing information systems for military purposes. This paper will give survey on recent projects explaining Macedonian approach related to this problem. In addition, the development of these solutions in SOA based purposes will be discussed.

Within the Macedonian e-Gov project (2004-2011) [8], different services have been developed within different information systems. These solutions have increased efficiency and transparency in specific public sectors, but problems appear when the interoperability of such services has to be established.

The following information systems: Information System Documentum; Information System eParliament; Information Border Management System (IBMS) and Interoperability System, can be considered as related to IIS.

All these Information Systems are SOA-based. As a result, information integration between IIS and selected Information Systems should be simple.

The goal of the **Information System Documentum** is to manage and store documents with different functions in a proper and convenient way. This system's consumers are:the government of the Republic of Macedonia; The Ministries; The General Secretariat; The Parliament; The Justice Secretariat; The Euro-Atlantic Integration Secretariat; The Ohrid Framework Agreement Secretariat; The Administration service.

Information System eParliament solution refers to the interior judicial processes. It is responsible for the daily coordination in decisions making which is derived from the judicial processes in Parliament. At the same time, this Information System integrates the Parliament and different institutions that are involved in the decision making.

The **Integrated Border Management System – IBMS** provides a platform for Information sharing, controlling and monitoring the state border. This system should provide coordinated information sharing between the state authorities that are responsible for the border management and security.

ISBN: 978-1-61208-152-6

Important information for the IIS contained in the IBMS is:State entry or exit of people, goods and vehicles; Detecting organized crime; Monitoring potential smuggling across the borders; Checking and monitoring the transfer of materials and infectious diseases, etc.

Data sharing between these systems has to be fast and efficient. The **Interoperability System should** help avoiding data duplicates and reduce errors caused by different systems' inconsistencies.

Technologically advanced countries use SOA based information systems in military domain. The main reason for this is increased level of security. Medlow [9] explains the interest about deploying SOA in ICT systems which are part of the military and civilian domains. SOA implementation in military domains of the systems used by land forces, peacekeeping or other kind of operations as a part of multinational contingents (led by NATO, UN or EU) is usually a big challenge.

Anschuetz [10] shows that the SOA implementation in the systems can be used as a possible solution for reorganization and optimization of the business processes which affords platform's independent application usage and implementation and its integration into organizational infrastructure. The usage of SOA offers benefits in terms of service interaction with external clients. As an example, the usage of SOA in a chain of supplying partners and other clients can be indicated. New generations of applications joined with integration processes of automation, business analysis and information integration allow the operators to extend the SOA benefits. In addition, recent researches related to this issue show that various information systems are created with emphasis on the evolution of business processes and information toward SOA that can be used to design suitable solutions for unsafe missions and network centric operations.

Bruce [11] explains that authorities responsible for building defense systems recognize the neediness of creating a full service oriented platform which will be used for operational military applications and services. As examples, the concepts for Network Core Services (Department of Defense (DoD)) and Network enabled core services (NATO) can be mentioned. In the same framework, the Australian Department of Defence suggest similar architecture (Single Information Environment Architectural Intent 2010). The IBM SOA Foundation (Triton Core) integrates different software products, best practice and consists of pattern which provides the elements required to implement and integrate SOA [12], [13] in organizational infrastructure [14], without financial implication caused by the coding and modifying processes.

Triton Core System consists of Enterprise Service Bus (ESB) and core for services [15], which is connecting the applications and resources of the other services in order to establish "Net-Centric" solution. Generally, the newly created Triton Core System has decreased the costs of Australian Defense Department and Australian defense industry. This solution raises the efficiency toward Single Information Environment architecture.

Mechling in [16] is describing the usage of the SOA by the Finnish Defense Forces (FDF). FDF are requested to increase their membership in the coalition with various specifics of military and civilian organization. Critical scenario which should be expected during their activities is coordination between Air Force, Navy, Police, Hospitals, and other military and civil groups. Technological incompatibility can cause coordination complications in the moment when groups use different technological architectures and communication protocols. FDF C4 (command, control, communications and computing) systems were created to support military domain, but these systems are „stove-piped" for supporting land forces, Navy and Air Force operations in the same time. In order to avoid this potential obstacle, the Program FINED was developed. The main reason for developing this program is implementation of the SOA for increased efficiency and creating reusable technology.

Radcliffe [17] is explaining that command and control (C2) information systems exploited in headquarters and in an operational level are using SOA. In order to increase capability of information sharing in military environment, the SOA approach allows flexible increase of the capability to share information through integration and systems interoperability based on commercial-off-the shelf (COTS) technology and standards.

This paper will give contribution in usage of SOA for developing prototype of IIS, which fulfills the requirements for intelligence disciplines as an Imagery Intelligence (IMINT), Signals Intelligence (SIGINT), Measurement and Signature Intelligence (MASINT), Electronic Intelligence (ELINT), Open-source Intelligence (OSINT), Human Intelligence (HUMINT) that answer to the requirements for intelligence cycle on which contemporary model of the Macedonian intelligence should be based.

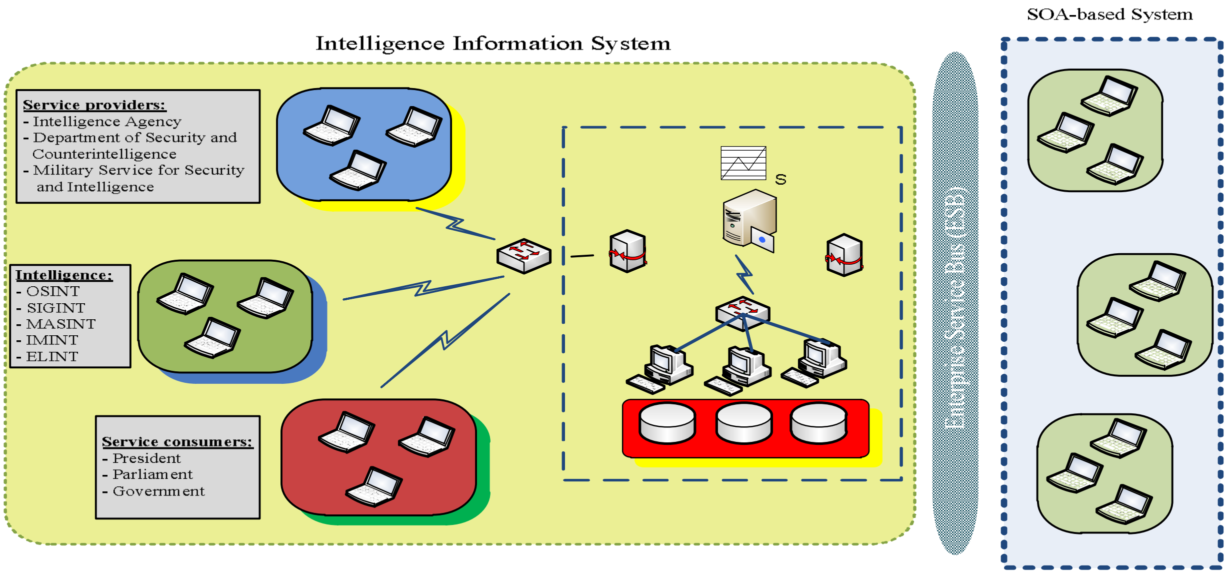


Figure 1. Model of SOA-based Intelligence Information System

III. IIS MODEL

The Model of SOA-based Intelligence Information System shown on Figure 1 should completely fulfill the intelligence role and assignments and its connection to the information systems of similar institutions.

Presented IIS model shows three types of users: service providers, service consumers and Intelligence. Service consumers are institutions (Crisis Management Center, MOI, Intelligence Agency or others) that have the need to get information from IIS or to give information as a notification.

According to security procedures service providers provide the consumers with required information.

Intelligence is based on several intelligence disciplines: IMINT, SIGINT, MASINT, OSINT, etc. In order to fulfill the requirements of intelligence disciplines, several tasks should be completed; gathering information (assessments, analyses, generating reports, etc.), then its verification and notification (e.g., political and security situation in foreign countries related to security of investments), etc.

All services are getting information from appropriate service providers through Information systems of the government institutions or the agencies which are included in Intelligence cycle. In addition, it is possible other Information systems to be service provider for inter-institutional governance. Service providers among system support for workflow processes define web services which are exploited from the users with appropriate security level to service registers.

Our methodology of developing SOA-based Intelligence Information System consists of several postulates. Experiences from recent researches which refers to SOA show that agencies, departments, institutions and other stakeholders can push and pull data on a standardized and flexible manner through communication interfaces using XML schema and web services.

The first postulate defines data exchanging methodology within SOA. It should be compatible with publicly described solutions for information systems which are supporting intelligence functions.

The second postulate focuses on the usage of SOA for information systems design, with the intention of finding relevance for developing Intelligence IIS. SOA should be treated as a standard for reusing information that is loosely coupled. This postulate enables independency of the implementation platform provided that hardware and software replacement without negative implications toward other components of a system as far as communication interface of service is not changed.

The third postulate describes functionalities for system end users of the IIS. According to user's division, user's functionalities of IIS can be explored from different aspects. Intelligence, as a system end user of IIS, is based on intelligence disciplines that are divided on sub-intelligence disciplines. These disciplines have different implementation as services within different government institutions, departments or military force units as a component of national security systems (Intelligence Agency, Ministry of Interior (MOI), Ministry of Defense (MOD), Ministry for Foreign Affairs and others). Intelligence services can be divided in three categories: data entering, data verification and notification (assessment of services for certain country could vary depending of security policies of the country).

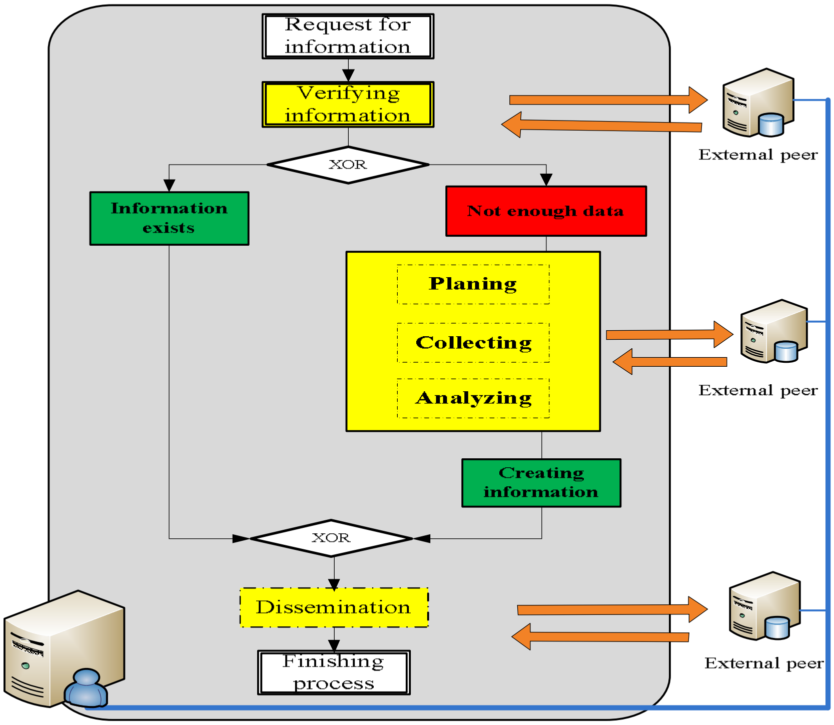


Figure 2. Example process of IIS [18]

The fourth postulate provides means that enable the future development of the IIS. In order to meet its future requirements, information infrastructure should be adoptable and flexible, and thus it needs to fully support information sharing process. The information sharing should be the basis for system development. It can help the authorities in decision making process, enabling them to plan actions in a convenient way. In order to accomplish this, one should define the model of information system integration which is undependable of the unique technologies and integration platforms.

The fifth postulate suggests that security standards needed to be implemented in order to achieve certain level of security. SOA-based information systems must be protected from intrusions and other vulnerabilities. Term security in these circumstances indicates establishing mechanisms which have system protection functions. In the phase of system designing, possible attacks and threats should be explored and according to them system protection mechanism should be designed. Usual vulnerabilities related to information system security refer to message interception, changing the context of message, denial of service, access denied, etc.

These five postulates can be used as the basis for SOA system deployment on the top of appropriate IT infrastructure. Such IIS achieves minimum requirements for designing services that are needed to be implemented in intelligence process with internal functions which can be processed from external IIS peer [18].

IV. Example Process with Intelligence Information System

In this chapter we will demonstrate the function of distributed IIS that follows suggested postulates. Figure 2 presents typical example process of request for intelligence information that should be established in intelligence sectors departments or agencies. This process has three functions which should be performed from external nodes in the network [18].

Function 1: As the client issues requests for certain information, this request ought to be accepted without exception if there is information available. In the case when there is no available information the request should be treated as request for creation of such information by certain agency that will be delegated to be information provider. The necessary inputs of data that have to be sent to external peer include the form request for information. According to this, this function executes computation steps in order to check whether information can be sent directly to the client who requested the information or additional data would be needed for creation of the information related to request. At the end, external functions send information messages back for local process.

Function 2: If there is not sufficient data for creation of intelligence information, a second function is started and its basic role is to decide which data is necessary or critical for creating intelligence information. Input values for this function is data from received request for

ISBN: 978-1-61208-152-6

intelligence information, followed by the information of intelligence disciplines which should be used for collecting information about which department, agency or sector is responsible for every part of process. In that manner this function is starting information planning, then collection and later analyses of collected information. In order to simplify this process, one can suppose that an agency which is responsible for producing intelligence information should deliver information in time. The value of the function in this case is a complex object and it consists of data related to intelligence information and expired date for their usage.

Function 3: When there is enough data related to Intelligence in a database of stakeholders, information creation can be started. The third function creates data records to whom intelligence information are sent explaining how long the information is valuable, etc. All data is wrapped as a complex object and it is sent to node of IIS which is the requestor for intelligence information.

Local peer in this example wraps requested data as objects in which the remote function calls are. Local database of intelligence stakeholder receives complex return objects and record them as a second master data.

This example shows how single functions of a process for requesting intelligence information is done on external IIS peers and how a local IIS system benefits from using external business logic, e.g., by using optimization functions.

V. Expected Achievments

Expected achievements that refer to SOA concept for developing intelligence information system are the following [19]:

1. Proposed model of IIS fulfills requirements for intelligence disciplines, and answers to requirements for intelligence cycle on which the contemporary model of Macedonian intelligence should be based.
2. Proposed approach for application integration which shows how selected information systems from Macedonian e-Gov project should exchange information.
3. Proposed approach describes which describes how to avoid technology dependence for the creation of information systems completely based on SOA.
4. Proposed approach which suggests that SOA security standards should be used in order to achieve appropriate level of access control and authentication.
5. Proposed model which integrates protocol for the information's distributed searching (online or near-line) depending on situation's importance in order to make the senior decision-makers an appropriate decisions.

Conclusion

Intelligence Information System Model gives contribution in Homeland Security and Civil Military Emerging Risks assessment through the possibility of providing information in the appropriate way by implementing pushing and pulling mechanisms into information systems, then by selection of data and creation of information from raw data, that can be used in creating intelligence products and dissemination reports to the authorities. In our case, this is done by IIS based on SOA which follows the five postulates that enables flexible and secure design of IIS.

References

[1] Air Combat Command, - Version 2, CONOPS UAV, Section 6 - Communication Integration and Interoperability, http://www.fas.org/irp/doddir/usaf, US Air Force; 3 Dec 1996

[2] P. Baglietto , M. Maresca, et al.. "Stepwise deployment methodology of a service oriented architecture for business communities." Journal: Information and Software Technology 47(6), pp. 427-436. 2005

[3] R. R Burk.. "Enabling Citizen-Centered Electronic Government" 2005-2006 Action Plan. USA. Office of EGovernment and Information Technology. 2005

[4] M. H. Burstein "Dynamic Invocation of Semantic Web Services That Use Unfamiliar Ontologies." IEEE INTELLIGENT SYSTEMS (JULY/AUGUST). (vol. 19 no. 4) pp. 67-73. 2004

[5] M. Castellano "An e-Government Cooperative Framework for Government Agencies". 38th Hawaii International Conference on System Sciences. Hawaii, IEEE. pp. 121c-121c. 2005

[6] T. Erl, Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Published by Prentice Hall. 2004

[7] "Exploring New Command and Control Concepts and Capabilities Final Report", NATO SAS-050, January 2006

[8] e-Gov Project - Paves the way for modern Macedonia, http://www.egov.org.mk, consulted of January 2011

[9] D. MedlowSaab Systems, "Extending Service Orientated Architectures to the Deployed Land Environment", Military Communications and Information Systems Conference (MilCis) Australia, http://www.milcis.com.au/, 2009

[10] B. Anschuetz, *Director*, SOA WW Business Development & Client Engagement Management, IBM Software Group, "Service Benefits – Life Beyond SOA", Military Communications and Information Systems Conference(MilCis) Australia, http://www.milcis.com.au/, 2009

[11] M. Bruce and A. Heys, IBM, "Introducing the Triton SOA Foundation for Military Systems Integrators and Developers", Military Communications and Information Systems Conference(MilCis) Australia,

http://www.milcis.com.au/, 2010

[12] M.P. Papazoglou. "Extending the Service Oriented Architecture". Business Integration Journal, pp. 18-21. February 2005

[13] S. Kumar and R. Rana. "Service on demand portals: A primer on federated portals". Web Logic Developers Journal:WLDJ, pp. 22-24. September/October 2004

[14] D. Krafzig, K. Banke, and D. Slama. “Enterprise SOA:Service Oriented Architecture Best Practices”. Prentice Hall, 2005

[15] A. Anagol-Subbaro. “J2EE Web Services on BEA WebLogic.” Prentice Hall, Upper Saddle River, Newy Jersey, 2005

[16] J. Mechling, Lecturer in Public Policy,”Finnish Defense Forces – Network-Centric Operations”, John F. Kennedy School of Governance, Harvard University, 2007

[17] S. Radcliffe, L. Trotman, and H. Duncan “Supporting Capability Evolution Using a Service Oriented Architecture Approach in a Military Command and Control Information System”, http://nectise.com/pdfs/2_Stewart%20Radcliffe.pdf

[18] N. Brehm , J.M. Gómez,: Secure Web Service-based resource sharing in ERP networks. International Journal on Information Privacy and Security (JIPS) 1 pp. 29-48, 2005

[19] J. Achkoski, V. Trajkovik , and M. Dojcinovski “SOA Approach in Prototype of Intelligence Information System” ICT Innovations 2010, Web Proceedings, ISSN 1857-7288 pp. 149-160. 2010

Switching IT Outsourcing Providers – a Conceptual Framework and Initial Assessment of Critical Success Factors

Matthias Olzmann
Business Solutions
noventum consulting
Muenster, Germany
e-mail: Matthias.olzmann@noventum.de

Martin Wynn
Department of Computing
University of Gloucestershire
Cheltenham, UK
e-mail: MWynn@glos.ac.uk

***Abstract*—Although IT outsourcing is a growing industry and a common topic in the literature, there is limited research which critically analyses and assesses the switching of IT outsourcing providers – in particular the factors contributing to success are under-researched. This article explores this growing area of management and consultancy activity by analyzing the existing literature in the field. This allows the identification of critical success factors that are pertinent to the switching of providers and the development of a conceptual framework for further research.**

Keywords - service providers; outsourcing; IT outsourcing; ITO; switching providers; critical success factors.

I. Introduction

When companies outsource their IT for the first time, it can be assumed that the majority of IT experts will transfer from the client company to the IT outsourcing (ITO) provider. Together with the IT experts, the client specific knowledge is transitioned to the provider. This reduces the negative performance impact. In contrast, when providers are switched, it cannot be anticipated that the majority of IT experts (together with the client specific knowledge) will transition from the incumbent provider to the new provider.

It can be assumed that the leaving provider has only marginal interest in actively supporting the incoming provider, for example with knowledge transition. This results in major challenges for the tripartite relationship (client, incumbent provider, new provider).

A main building block in switching ITO providers is the transition. Transition is a complex, risky, and challenging building block of strategic importance which begins after the contract is signed and ends with service delivery. Two thirds of all issues can be tracked to the transition [1] [2].

Despite growing interest in topics such as sourcing the IT back in-house or switching providers [3] [4] [5], no studies have holistically focused on how successful ITO transitions are performed for clients switching service providers.

The factors contributing to a successful transition from the incumbent provider to the new provider are not fully understood. Yet understanding the factors contributing to a successful transition is vitally important. For the client, these factors determine on the one hand the success or the failure of the whole outsourcing endeavour; and on the other hand, ultimately the survival of the overall business as it is linked to the successful switch of the ITO providers. This article sets out to review available literature relating to this topic and draw conclusions regarding critical success factors for achieving the switching of service providers. In the following section, a wide range of literature relating to ITO is systematically reviewed. This leads to a discussion of critical success factors in section three, focusing on both the pre-delivery phase and the critical transition process. Section four then makes some concluding remarks relating to the analysis of existing literature, and highlights a conceptual framework for future work in this field.

II. Initial Literature Review

“Outsourcing can be defined as turning over all or part of an organizational activity to an outside vendor” [6]. In contrast to other types of outsourcing, ITO affects the complete organisation – IT “is pervasive throughout the organization” [7]. Reference [4] suggests that in an ITO deal, the IT is either partly or fully turned over to “…one or more external service providers”.

A. ITO History and Market Development

Even though large scale modern ITO began in 1989 with the Kodak outsourcing deal [4], some researchers argue that ITO “is still at the early stages of the profession itself” [8]. Kodak was not the first ITO deal in history although other deals had only received scarce attention. “It was not until Kathy Hudson, the Kodak CIO, announced to the world that Kodak had entered into a ‘strategic alliance’ with its IS partners, led by IBM but also including DEC and Businessland, did the world sit up and take notice” [7].

Many scholars and practitioners forecast further growth of the ITO market [8] [9] [10]. Reference [11] emphasises that: “on conservative estimates, looking across a range of reports and studies, global ITO revenues probably exceeded $270 billion in 2010; it is very clear that, with its 20-year history, outsourcing of IT and business services is moving into becoming an almost routine part of management, representing in many major corporations and government agencies the greater percentage of their IT expenditure”. All reports (Gartner, Everest, NASSCOM, and IDC) reviewed have indicated a global growth of ITO in the range of 5-8% per year [11].

B. *Reasons for ITO*

Research findings indicate that the main reasons for ITO are driven by the goal of cost reduction, the focus on core capabilities and a desire to access resources of the provider such as superior capabilities, expertise and technology [6] [11]. The primary reason for outsourcing in 90 % of the reviewed literature indicated the motivation of cost reduction [11]; but not all researchers agree that the goal of cost reduction and performance improvement will automatically be achieved - no matter how the outsourcing endeavour is managed. Reference [6] argues that "this overly optimistic view of outsourcing derives from the fact that most articles about outsourcing are written during the so called 'honeymoon' period i.e., just before or after the contract is signed". Outsourcing strategies therefore need to be deliberate to increase the companies' overall performance.

From the perspective of the ITO provider, *long-term revenue* is the primary reason to enter outsourcing arrangements. Reference [7] points out that "long-term outsourcing arrangements help stabilize vendor business volume and revenue, making planning more predictable, and increase shareholder's comfort levels".

The typical length of ITO contracts is generally 5-10 years and "thus, both client and vendor have come to expect that during the life of the contract, some form of renegotiations will be likely" [7]. The rapid growth and the complex nature of ITO have not been without impact. Recently a number of outsourcing deals have experienced both serious problems and the premature discontinuation of contracts [3] [4] [5] [7] [10]. This leads companies to re-consider sourcing options and strategies. The discontinuation of contracts results in several strategic options. Regarding ITO contracts, "as much as 50%" of these are ended for other options such as switching the provider, or IT backsourcing [4]. Other researchers have found that most clients stay with the incumbent provider [8] [10]. Reference [10] estimates that 25% of contracts will be awarded to new providers and merely 10 % will be back-sourced. Reference [3] notes the reasons for changing ITO providers as follows:

- "Dynamic changes in the customer landscape (e.g. the client organization may have outgrown the supplier)
- A shift in management's risk tolerance
- Changes in the supply market (e.g., emergence of new or specialized players)
- Supplier rationalization (e.g., consolidation to enhance bargaining power)".

C. *Factors Influencing Sourcing Options*

What factors influence sourcing option decisions when contracts are re-evaluated? Switching costs play a vital role in sourcing decisions – they are a good indicator for understanding and predicting clients' outsourcing decisions after re-evaluating sourcing options [12]. Reference [4] argues that "the greater the information transfer/setup costs, the more likely that outsourcing continuation will be the strategic choice, vendor switching will be the intermediate choice, and backsourcing will be avoided". The researchers warn that "high switching costs might entrap the customer organization into a 'no change situation', forcing it to continue outsourcing IT work to the same vendor". Although customer entrapment has been noted - not much has been written in the academic literature about how to avoid or adequately address it.

In contrast to high switching costs, if companies anticipate low switching costs and the option to choose from many vendors, there is "no real advantage in recontracting with the same vendor" [6]. Despite the significance of switching costs, the measurement of these costs remains unclear [12].

A study analyzing the influencing factors of sourcing options found that firms which decided to switch providers or to backsource typically experienced high service quality and low relationship quality. They acknowledged that "relationship quality plays on important role in the decision to switch vendors. Of our three groups, those that switched vendors had the lowest perception of trust, commitment, culture, and communication in relation to their vendors…hence, the building of trust between an outsourcer and a firm is far more a socio-emotional condition than it is a matter of providing excellent product and/or service" [5].

The importance of relationship for staying with the current provider has been highlighted in a previous study [6], where the researchers found a high interest in staying with the same provider if relationship specific investments have been made. The risk of losing knowledge and the potential service operation distortions prevents companies from switching ITO providers. Reference [13] argues that the "switching of IT vendors is seen to impose too much short-term operational risk to justify the financial savings and quality improvements that could accrue from a relationship with a new vendor".

D. *ITO Success*

ITO success has not been extensively researched and there are contrasting conclusions on the contributing success factors. It is not clear if this is due to the lack of a generally accepted construct of a success definition or because "ITO success is so idiosyncratic that one must assess it against each organization's own, different criteria" [14]. Reference [7], in a widely cited (more than 500 times according to Google scholar) literature survey and analysis, notes that "outsourcing success is usually viewed as the attainment of economic, technological or business-related benefits. Satisfaction with the benefits attained is often used as an indicator of outsourcing success".

Companies outsource their IT for different reasons, as previously noted. For example one company outsources to gain access to superior IT capabilities, another to focus on core competences, and another to reduce costs. This means that outsourcing success is dependent on the overall context. Thus, it is plausible that "any attempt to assess ITO success in terms of more detailed criteria, such as cost savings or focusing on core business, requires identification of the different criteria relevant to each organization for each different contract at the time of the study"[14].

Therefore it appears to be important to define factors contributing to outsourcing success before the contract is signed [15]. Reference [14] argues that success should be assessed by:

1. Defining most important outcomes before they actually materialise during the lifecycle of the contract
2. Measuring the extent to which the outcomes have been achieved.

Can outsourcing be considered as a standardised activity of everyday management with readily defined solutions? Reference [11] disputes this and concludes that "our review of 20 years of research establishes the common denominator that, for management and operational staff, outsourcing is far from easy". Reference [16] found that even skilled organizations don't work in a proactive mode and are hurt by slow organizational learning. Therefore, in order to reduce learning curves, it is important to understand how success can be defined and what the contributing factors are. Reference [15] suggests a more abstract description of success factor such as:

- "Use 'best outsourcing practices' as major references for corporate outsourcing decision.
- Clearly understand the goals, objective, scope, budget, and the duration of IS outsourcing project….
- Select a reputable vendor and then communicate well on the corporate outsourcing plan.
- Realize the legal issues related to contract negotiations and signing.
- Communicate well with employees and stakeholders about the outsourcing plan; this may reduce the severity of resistance."

Even though these factors are useful to get an overview about common success factors, they are of limited applicability for the specific issue of switching ITO providers. A review of 191 ITO articles relevant to practice from the early 1990s until 2009 found that "the three major categories of determinants of ITO success are *ITO decisions, contractual governance, and relational governance*" [17]. These determinants are depicted as direct relationships to ITO success in Fig. 1.

Although organizational capabilities are also important as a success contributing factor, they are neither depicted in Fig. 1 nor are they described in the section about the determinants of success. Reference [17] recognises that "the most widely cited papers on this topic identify a mix of complementary capabilities that lead to ITO success". Reference [18] develops this further into a list of nine pertinent organizational capabilities shown in Table I.

Reference [17] summarises research findings thus: "overall, we know *ITO decisions* that entailed selective use of outsourcing, the involvement of senior managers, and rigorous evaluation processes, were associated with higher levels of ITO success. *Contractual governance* also positively affected ITO success. In general, more contract detail, shorter-term contracts, and higher-dollar valued contracts were positively related to outsourcing success…. *Relational governance* positively affected ITO outcomes. Trust, norms, open communication, open sharing of information, mutual dependency and cooperation were always associated with higher levels of ITO success". The researchers found that top management commitment/support is the most critical success factor and that trust plays a vital role in the success of ITOs. Reference [7] adds that "Sabherwal also suggests that a 'psychological contract' exists in outsourcing relationships. This contract, which consists of unwritten and often unspoken expectations, is supported by the level of trust between the parties, and plays a role in resolving unanticipated problems or changes in the accomplishment of outsourced activities".

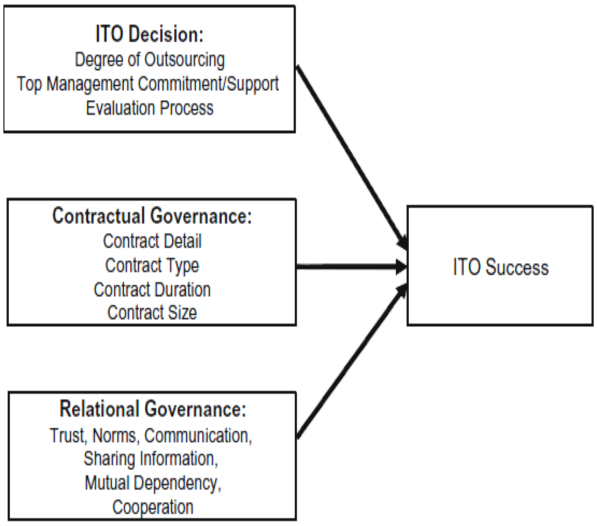


Figure 1. Three main categories of determinants of ITO success [17]

Based on these findings, it seems clear that trust and the management of relationships between the client and the outsourcing provider are important factors contributing to success. However, given that significant amounts of capital are often invested in outsourcing deals, clients should probably not solely rely on relational governance factors such as trust and relationship. Reference [6] endorses this view in asserting that it is not advisable to completely rely on partnership factors and neglect contract negotiation – "a

TABLE I. ORGANISATIONAL CAPABILITIES RELEVANT TO ITO SUCCESS [18]

	Capability		Capability
1	IS/IT leadership	6	Informed buying
2	Business systems thinking	7	Contract facilitation
3	Relationship building	8	Contract monitoring
4	Architecture planning	9	Vendor development
5	Making technology work		

good contract is essential to outsourcing success because the contract helps establish a balance of power between the client and the vendor".

Understanding the budget is of critical importance [15]. Reference [6] proposed the hiring of external experts as they know the hazards of outsourcing and how they can be managed. They argue that the additional costs may be justified in relation to the potential impact of the hidden costs. Other researchers found that "managing costs is less important than managing portfolio configuration, complexity and risk" [16]. This implies the importance of actively managing the outsourcing provider.

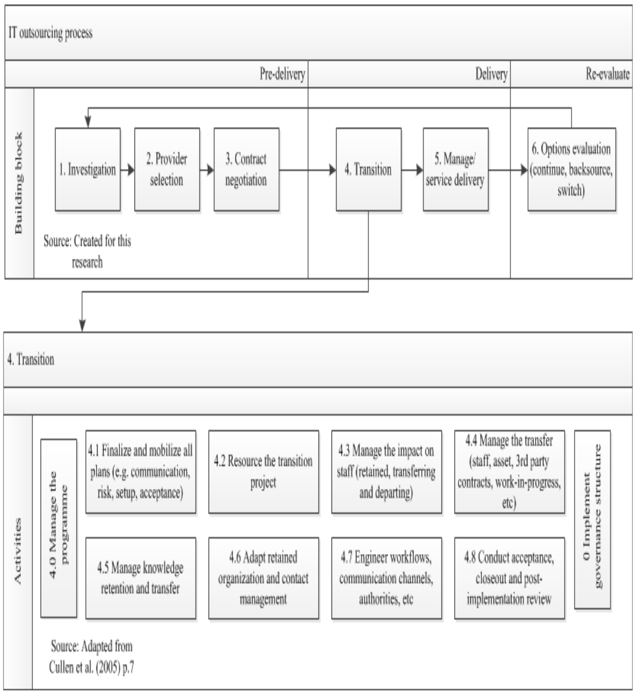
Success itself can be considered an important factor contributing to success. "ITO success fuelled higher levels of trust (relational governance, built stronger client and supplier capabilities, and determined the kinds of ITO decisions and ITO contracts clients made moving….Conversely, ITO failure fuelled greater need for controls, monitoring mechanisms, tougher contracts, and determined the kinds of ITO decisions clients made" [17].

E. ITO Methodologies

Reference [16] defines a detailed process model using nine building blocks with 54 activities. This model describes the complete ITO process lifecycle and appears to be the most comprehensive in the academic literature. Many ITO process models distinguish between activities before signing the contract (pre-delivery) and after signing the contract (delivery & re-evaluate) [3] [13] [16] [19]. The ITO process model for this research is depicted in Fig. 2. The six major building blocks are: investigation, provider selection, contract negotiation, transition, manage/service delivery, and options evaluation. The first three building blocks can be considered as pre-delivery phase, the next two can be considered as delivery-phase, and the last activity can be considered as the re-evaluation phase.

Transition is a complex, risky, and challenging building block of strategic importance which begins after the contract is signed and ends with service delivery. Transition "sets the tone for the entire relationship and involves handover of outsourced services from either the client's internal IT department or the incumbent service provider" [1]. Transition can be summarized as "a pre-requisite to implementing an outsourcing contract successfully" [1]. Reference [20] defines the transition stage as "implementing the new way of operating" and states that it is the goal of transition to ensure that the new way of working is realized.

Transition includes the following activities: "conducting knowledge transfer, determining and implementing new governance structures, and applying the processes of the service provider" [1]. This demonstrates that many actions need to take place during transition before "an outsourcing project can be actually implemented" [15]. Reference [16] has identified the main transitional activities as shown in Fig. 3. The cost the transitional building block can take a significant portion of the overall costs, with some researchers suggesting that the cost of transition ranges from 2 to 15% of the total cost of the first year of the outsourcing deal [1]. It is assumed that "over two-thirds of the problems in these unsuccessful engagements arise due to failed or poor transition" [1]. Due to the lack of statistical information regarding what percentage of switching ITO providers fail due to poor transition, it is assumed in this review that the percentage is at least as high as this.


IT outsourcing process
Pre-delivery
Delivery
Re-evaluate
Building block
1. Investigation
2. Provider selection
3. Contract negotiation
4. Transition
5. Manage/ service delivery
6. Options evaluation (continue, backsource, switch)
4. Transition
Activities
4.0 Manage the programme
4.1 Finalize and mobilize all plans (e.g. communication, risk, setup, acceptance)
4.2 Resource the transition project
4.3 Manage the impact on staff (retained, transferring and departing)
4.4 Manage the transfer (staff, asset, 3rd party contracts, work-in-progress, etc)
0 Implement governance structure
4.5 Manage knowledge retention and transfer
4.6 Adapt retained organization and contact management
4.7 Engineer workflows, communication channels, authorities, etc
4.8 Conduct acceptance, closeout and post-implementation review

Source: Created for this research

Source: Adapted from Cullen et al. (2005) p.7

Figure 2. ITO process with the focus on transition

III. CRITICAL ISSUES IN SWITCHING PROVIDERS

When companies outsource their IT the first time it can be assumed that the majority of IT experts will transfer from the client company to ITO provider. Together with the IT experts, the client specific knowledge is transitioned to the provider. This reduces the negative performance impact. In contrast, when providers are switched it cannot be anticipated that the majority of IT experts (together with the client specific knowledge) will transition from the incumbent provider to the new provider. Reference [13] concludes that "a long-term outsourcing relationship with a prior vendor means that much daily operational knowledge stays with the prior vendor. The client's knowledge loss exacerbates the problem of knowledge transfer as the client no longer possesses the information that the new vendor critically needs to service the client". The new provider requires close cooperation with the incumbent provider, who can pursue two different exit strategies. They can either actively co-operate with the new provider or "pursue a hostile strategy of being uncooperative" [21].

It can be assumed that the leaving provider has only marginal interest in actively supporting the incoming provider, for example with knowledge transition. This is

ISBN: 978-1-61208-152-6

particularly the case if the outgoing provider is not contractually obliged to support the incoming provider. Reference [21] named source code as an example of this, but the findings apply to all client specific knowledge.

Reference [6] found that "many managers are reluctant to anticipate the end of an outsourcing contract. Therefore, they often fail to plan an exit strategy". With the risk of loss of knowledge comes the risk of degraded service quality. Reference [13] found that switching often lead to "temporary service disruptions of operations, lowered service levels and frustrations and dissatisfaction among the client employees". In addition this can lead to broken transition milestones, extended project duration and additional costs. Clients should take into consideration that once the contract of the incumbent provider has expired, the provider will leave regardless of whether the new provider is already prepared to deliver the service [21]. This can negatively impact service levels and even risk business continuity if the new provider is not completely ready. Alternatively, the client needs to be prepared to additionally pay the old provider for extending the contract until the new provider can adequately deliver the IT services.

When providers are switched transitional activities can be extensively resource draining for client, who needs to manage (monitor and correct) the operations of both the incumbent and new provider and additionally the transition between the two. Even relatively simple transitions where the IT can be transferred directly from the client to the outsourcing provider can be a costly phase and "in some cases, they (the transition activities) halved or even cancelled out the company's potential savings from outsourcing" [22]. It can be assumed that the transitional activities for switching providers are even more costly. As a general rule it can be stated that the more idiosyncratic the IT service to be outsourced, the more complex and costly the transition.

If the perception is that ITO can be handled as a commodity, there is a risk that companies which have chosen to switch outsourcing providers underestimate the effort, complexities and risks involved. Reference [3] has disputed the common perception that "once part of a business process has been outsourced, it can, if necessary, easily be 'un-plugged' from one supplier and 're-plugged' into another".

A. Pre-delivery Phase – Factors Contributing to Switching Success

The client should ensure that the new potential ITO provider conducts an extensive *due diligence* review. "Before the service providers make a final offer during contractual negotiations, a thorough due diligence activity is required to closely understand the actual outsourced work and its related dependencies." [1]. Due diligence is even more important when providers are switched to ensure that the interdependencies between client and leaving provider are fully understood. Due diligence lays the baseline for the overall project management of the outsourced activity, encompassing scope, time and quality definition. Reference [3] has noted the importance of identifying essential specific knowledge before the actual transition phase to avoid disruptions during transition.

However, it is questionable whether some clients have sufficient resources to successfully accomplish such preparations. Identifying knowledge gaps before the transition is likely to be only partly successful. Reference [13] noted that "at the time of the contract negotiations, both parties (client & new provider) were still largely unaware of the gaps in the knowledge that would trouble the change-over from the prior provider to the new provider". Much of the operational knowledge is only visible to the people involved in everyday operations. This means that the client and the new provider can possible face unexpected knowledge gaps during transition.

B. Building Block Transition - Factors Contributing to Switching Success

Good project management and realistic time schedules are critical. "Unrealistic transition timetables are a frequent source of trouble. Both buyers and providers should look with a sceptical eye at the viability of their transition timeframes" [2]. It is also important to incorporate project buffers or contingencies into the project plan. "Any organization that explores a new sourcing option in terms of suppliers, new services, or new engagement models…must plan on false starts. Executives often manage learning by pilot testing new sourcing options" [17]. Although this is a good method of learning and getting the experience for some sourcing options in principle, it is not easy to pilot test switching ITO providers in practice.

To effectively manage the transition the client needs to set up an overall transition governance structure. Reference [1] asserts that "both client and service providers need to develop and implement an appropriate governance model for efficiently conducting day-to-day activities and for monitoring it at a higher level". The governance structure should define project roles and responsibilities such as the project joint steering committee. All parties (client, new provider and old provider) should be part of the joint steering committee. Part of the responsibilities of the joint steering committee is it to manage conflicts and to implement a joint transition program to plan, monitor, execute, and report on all transition switch deliverables and milestones.

Managing the complex tripartite relationship is resource intensive. Reference [3] emphasises the importance of sufficient resources from the client to manage the transition and materializing risks. The authors call for the active involvement of the client management to ensure that the old provider supports the new provider as needed and therefore minimize service disruptions.

Reference [13] found that: "switching required close collaboration and mutual adjustment among all parties". Although the motivation of the old ITO provider to support the new provider might be low, it is a critical success factor for the overall transition success. "An uncooperative old supplier or an insensitive new supplier increases the risk of transition problems. Organisations must therefore carefully manage the delicate tripartite relationship tensions" [3]. Reference [13] also found that the old supplier is often needed to develop joint knowledge together with the new supplier to ensure that all parties meet their responsibilities - “critical to the success is the transfer of the knowledge of the client’s environment and processes. Poor knowledge transfer may result in disruptions of operations, lowered service levels, and frustrations and dissatisfaction among the client’s and the new vendor’s employees”.

Reference [6] emphasises the importance of “commitment of employees transferred” to the provider and that the outsourcing success is related to it. “First, key employees must be retained and motivated. For most activities, outsourcing does not mean transferring all the employees to the vendor. When an activity has been performed in-house for a long period of time, firm-specific knowledge about how to run the activity smoothly has accumulated. Employees who possess this firm-specific knowledge must be identified”.

What does this mean for switching providers? Clients need to identify employees from the incumbent provider who possess important firm specific knowledge and try either to reintegrate them into the client company or make sure that they move over to the new client or ensure adequate knowledge transfer. However, it is likely that the leaving provider will block the transfer of personal to stay competitive [21]. Transferring key employees early to the new provider could negatively impact the production capability of the incumbent provider.

IV. Concluding Remarks

Even though the modern form of ITO practice effectively started in the late 1980s, it still cannot be considered a standardized routine management practice. Companies outsource their IT for different reasons though the primary objective is cost reduction. Several studies indicate a further growth of the ITO market of 5-8% per year [11]. The typical length of ITO contracts is 5-10 years [7] - a time span over which it is neither possible to foresee the clients’ IT requests nor to estimate the impact of the overall economic environment. Various factors have led a number of clients to cancel their contracts prematurely.

The options for clients are to continue with the incumbent provider, switch the provider, or IT backsource (i.e., in-source again). It is estimated that between 25% [10] and 50% [4] of clients do not continue the relationship with the same provider. Miscellaneous factors influence these three sourcing options, most importantly the anticipated switching costs, the relationship between client and provider, and the fear of losing knowledge.

ITO success has not been extensively researched and there are contrasting conclusions regarding the contributing success factors [14]. Research has found that success needs to be considered in the context of the specific outsourcing arrangement. Several academics agree that the desirable outcomes need to be defined before the ITO starts, and that outcomes should be systematically assessed after it has been finalized and is underway.

General ITO factors contributing to success can be grouped into the major categories of ITO decisions, contractual governance, relational governance, and organizational capabilities [17]. In the category of ITO decisions, top management commitment and support is the most important factor [17]. In the relational governance category, trust and relationship management play a vital role [17]. However, given that significant amounts of capital are often invested in ITO deals, clients should not completely rely on relational governance factors such as trust and relationship. Important capabilities are required for success such as cost control and provider management. In addition, success itself can be considered as an important factor contributing to success.

The outsourcing process may be conceptualized as six major building blocks - investigation, provider selection, contract negotiation, transition, manage/service delivery, and options evaluation. The first three building blocks can be considered as the pre-delivery phase, the next two can be considered as the delivery phase, and the last activity can be considered as the re-evaluation phase. The transition building block is a complex, risky, and challenging process of strategic importance which begins after the contract is signed and ends with service delivery. It is assumed that “over two-thirds of the problems in these unsuccessful engagements arise due to failed or poor transition” [1].

When providers are switched, it cannot be assumed that the accumulated IT expertise (both in terms of personnel and client specific knowledge) will transition from the incumbent provider to the new provider. This results in several major issues, which are significantly impacted by the strategy of the incumbent provider. Their reaction can be grouped into two categories – a cooperative strategy or hostile strategy. Clients are well advised to prepare for both scenarios. Switching providers can be extensively resource draining for clients, as clients need to manage (monitor and correct) the operation of the incumbent provider, the operations of the new provider and additionally the transition from the old to the new one. This means clients should budget and plan for extra resources and associated contingencies.

During the pre-delivery phase it is essential for a successful transition to identify specific knowledge that needs to be transferred. A strategy should be developed to establish how this knowledge will be transferred and key knowledge experts need to be identified. Clients may reckon that major knowledge gaps will only be recognised during the actual transition.

In the critical transition building block, several factors contributing to success have been identified. Conducting a stringent project management methodology with focus on realistic time schedules and incorporated buffers is an important ingredient for success. Implementing an effective governance structure plays a vital role for a successful transition when providers are switched. Ensuring early knowledge transfer and the transfer of key knowledge

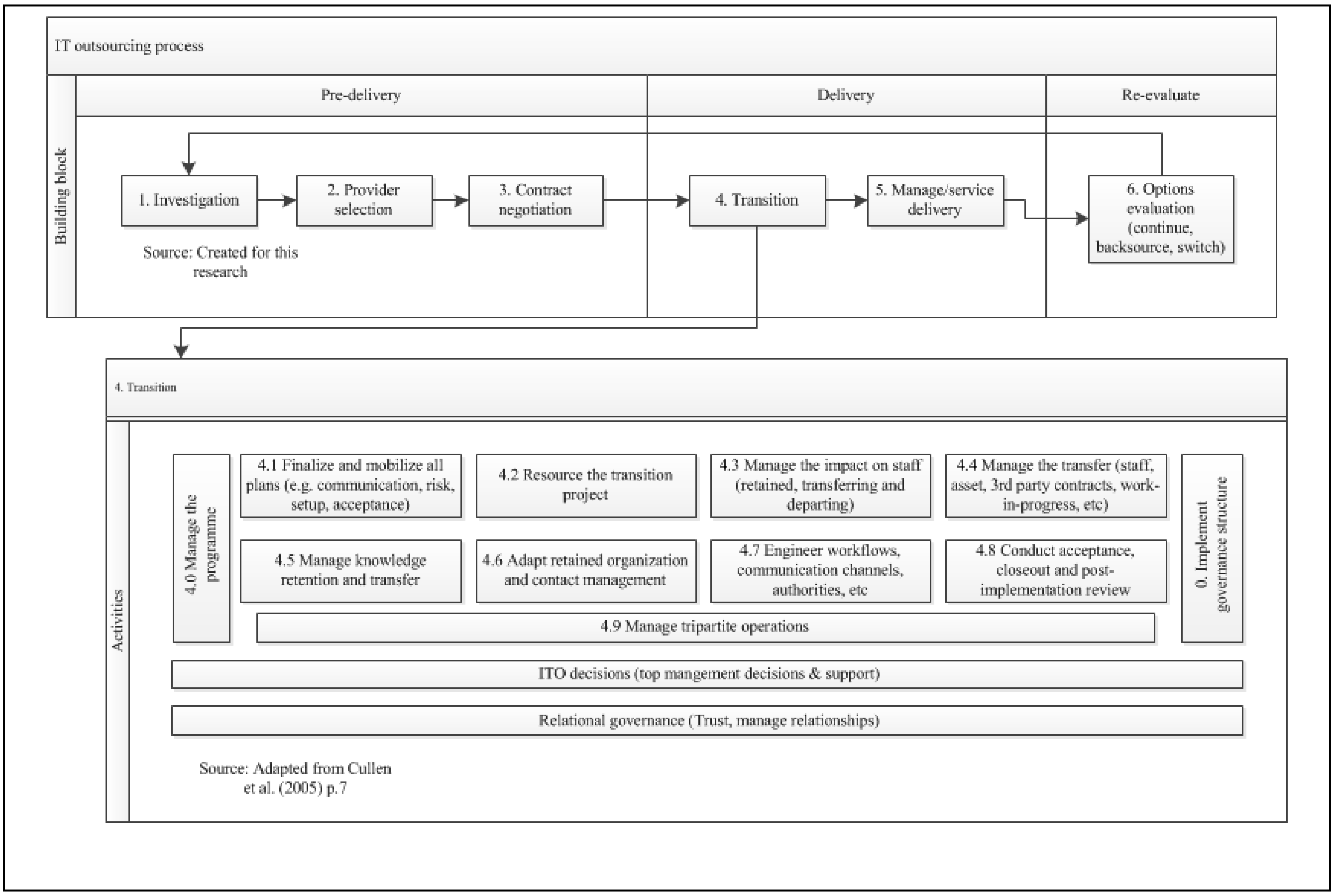

IT outsourcing process
Building block
Pre-delivery
Delivery
Re-evaluate
1. Investigation
2. Provider selection
3. Contract negotiation
4. Transition
5. Manage/service delivery
6. Options evaluation (continue, backsource, switch)
Source: Created for this research
4. Transition
Activities
4.0 Manage the programme
4.1 Finalize and mobilize all plans (e.g. communication, risk, setup, acceptance)
4.2 Resource the transition project
4.3 Manage the impact on staff (retained, transferring and departing)
4.4 Manage the transfer (staff, asset, 3rd party contracts, work-in-progress, etc)
4.5 Manage knowledge retention and transfer
4.6 Adapt retained organization and contact management
4.7 Engineer workflows, communication channels, authorities, etc
4.8 Conduct acceptance, closeout and post-implementation review
4.9 Manage tripartite operations
0. Implement governance structure
ITO decisions (top mangement decisions & support)
Relational governance (Trust, manage relationships)
Source: Adapted from Cullen et al. (2005) p.7

Figure 3. Conceptual framework - Switching providers with the focus on transition

experts from the incumbent provider are two of the most important factors for success. Finally, managing the complex tripartite relationship is resource intensive but an important factor for success. The conceptual framework depicted in Fig. 3 has been developed to guide further research.

In conclusion, the switching of ITO providers is a complex, risky and resource intensive endeavour with the transition stage being the major building block in a wider process. However, not much is known about methods, processes and strategies for switching ITO providers as most research has focused on the initial outsourcing [15] [20] [23]. It is intended that this literature review and analysis will provide a useful starting point for subsequent research into these areas, and this is being pursued by the authors through primary research involving a wide range of practitioners from ITO clients and ITO providers. The scope of this research will be a) large ITO deals with total contract value of more than €150 million and b) at least 2 IT-services (e.g. network services and server production services) which have been outsourced and need to be switched. The scope of this research will not be limited to any specific industry.

REFERENCES

[1] E. Beulen and V. Tiwari, “Parallel transitions in ITO: making it happen,” in Global Sourcing of Information Technology and Business Processes Vol. 55, I. Oshri & J. Kotlarsky (Eds.), Springer Berlin Heidelberg, 2010, pp. 55-68.

[2] M. Robinson and P. Iannone, “9 ways to avoid outsourcing failure, a three-part approach to maximizing the value of an ITO deal” retrieved 30.01.2011, from http://www.cio.com.au/article/205186/9_ways_avoid_outsourcing_failure/?pp=1&fp=4&fpid=15 2007.

[3] K. Sia Siew, K. Lim Wee, and K. P. Periasamy, “Switching ITO suppliers: enhancing transition readiness,” MIS Quarterly Executive, 9(1), 2010, pp. 23-33.

[4] D. Whitten, S. Chakrabarty, S., and R. Wakefield, “The strategic choice to continue outsourcing, switch vendors, or backsource: do switching costs matter?” Information & Management, 47(3), 2010, pp. 167-175. doi: DOI: 10.1016/j.im.2010.01.006.

[5] D. Whitten and D. Leidner, “Bringing IT back: an analysis of the decision to backsource or switch vendors,” Decision Sciences, 37(4), 2006, pp. 605-621.

[6] J. Barthélemy and D. Adsit, “The seven deadly sins of outsourcing (and executive commentary),” The Academy of Management Executive (1993-2005), 17(2), 2003, pp. 87-100.

[7] J. Dibbern, T. Goles, R. Hirschheim, and B. Jayatilaka, “Information systems outsourcing: a survey and analysis of the literature,” Data Base for Advances in Information Systems, 35(4), 2004, pp. 6-98.

ISBN: 978-1-61208-152-6

[8] L. Willcocks, “Machiavelli, management and outsourcing: still on the learning curve,” Strategic Outsourcing: An International Journal, 4(1), 2011, p. 13.

[9] M. Benaroch, Q. Dai, and R. Kauffman, “Should we go our own way? backsourcing flexibility in IT services contracts,” Journal of Management Information Systems, 26(4), 2010, pp. 317-358.

[10] M. C. Lacity, L. Willcocks, and J. Rottman, “Global outsourcing of back office services: lessons, trends, and enduring challenges,” Strategic Outsourcing: An International Journal, 1(1), 2008, pp. 13-34. doi: 10.1108/17538290810857457.

[11] M. C. Lacity, S. Khan, A. Yan, and L. P. Willcocks, “A review of the ITO empirical literature and future research directions,” Journal of Information Technology, 2010 doi: 10.1057/jit.2010.21.

[12] D. Whitten and R. L. Wakefield, “Measuring switching costs in ITO services,” The Journal of Strategic Information Systems, 15(3), 2006, pp. 219-248. doi: DOI: 10.1016/j.jsis.2005.11.002.

[13] M. Alaranta, S. L. Jarvenpaa, “Changing IT providers in public sector outsourcing: managing the loss of experiential knowledge,” in 43rd Hawaii International Conference on System Science, Hawaii, 2010, pp. 1-10, doi.ieeecomputersociety.org/10.1109/HICSS.2010.101.

[14] S. Cullen, P. Seddon, and L. Willcocks, “ITO success: A multi-dimensional, contextual perspective of outsourcing outcomes,” in Second information Systems Workshop on global Sourcing: Service, Knowledge and Innovation, Val d’Isere, France, 2008, pp. 1-38.

[15] D. C. Chou and A. Y. Chou, “Information systems outsourcing life cycle and risks analysis,” Computer Standards & Interfaces, 31(5), 2009, pp. 1036-1043. doi: DOI: 10.1016/j.csi.2008.09.032.

[16] S. Cullen, P. Seddon, and L. Willcocks, “Managing outsourcing: the life cycle imperative,” MIS Quarterly Executive, 4(1), 2005, pp. 229-246.

[17] M. C. Lacity, S. A. Khan, and L. P. Willcocks, “A review of the ITO literature: insights for practice,” The Journal of Strategic Information Systems, 18(3), 2009, pp. 130-146. doi: DOI: 10.1016/j.jsis.2009.06.002 .

[18] D. Feeny and L. P. Willcocks, “Core IS capabilities for exploiting information technology,” Sloan Management Review, 39(3), 1998, pp. 9-21.

[19] V. Tiwari, “Transition during offshore autsourcing: a process model,” ICIS 2009 Proceedings, 33, 2009.

[20] S. Cullen and L. P. Willcocks, “Intelligent ITO: eight building blocks to success,” Butterworth-Heinemann, 2003.

[21] C Chua, W. Lim, S. Sia, and C. Soh, “Threat-Balancing in Vendor Transition,” in 3rd International Research Workshop on Information Technology Project Management, Paris, France 2008, pp. 19-26.

[22] J. Barthelemy, “The hidden costs of ITO,” Sloan Management Review, 42(3), 2001, pp. 60-69.

[23] D. Whitten, “Adaptability in IT sourcing: the impact of switching costs,” in Global Sourcing of Information Technology and Business Processes Vol. 55, I. Oshri & J. Kotlarsky (Eds.), Springer Berlin Heidelberg, 2010, pp. 202-216, doi: 10.1007/978-3-642-15417-1_11.

Goal Refinement for Automated Service Discovery

Tomas Olsson, May Yee Chong, Björn Bjurling
Swedish Institute of Computer Science
164 29 Kista, Sweden
Email: {tol, may, bgb}@sics.se

Börje Ohlman
Ericsson Research AB
Färögatan 6, 164 80 Kista, Sweden
Email: borje.ohlman@ericsson.com

***Abstract*—An important prerequisite for service composition is a versatile and efficient service discovery mechanism. The trends in service computing currently point toward a veritable explosion in the number of services that are or will become available as components in compositions of new services—Cloud Computing and the 'Internet of Things' are just two examples of such new trends. We hypothesize that it will become critical to be able to filter the discovered services with respect to pre- and postconditions, as well as other semantic aspects such as relevance. We have studied the use of *goals* as a means for describing semantic aspects of services (e.g., their effects). By using goals, services can be described on any arbitrary and useful level of abstraction. By a goal refinement algorithm, goals can be used not only for describing services, but also for improving the performance of service discovery. In this paper, we describe the goal refinement algorithm and our approach to incorporating it into our service discovery machinery.**

***Keywords*-Service Discovery; Service Composition; Semantic Web Service; Goal Refinement.**

I. Introduction

One major trend of today is to transition old and new software into a service-oriented environment for reusing and sharing services across organizational boundaries. Key enabling technologies for this transition are web service technologies such as WSDL [1] and SOAP [2]. As the number of services placed online increases, methods for simplified and intelligent service discovery have become an important area of research [3][4]. Service discovery is an essential prerequisite of service composition, and it is paramount that the service discovery mechanism can determine whether a service is relevant to the requirements in a service composition specification.

One avenue of approach to finding a solution to the discovery and composition problem is to use semantic web services where the services are described using ontologies [5][6]. A step toward this is the Web Service Modeling Ontology (WSMO) [5] and the corresponding Web Service Modeling eXecution (WSMX) framework [7]. Another similar ontology for modeling web services is OWL-S [6]. In particular interest for our work is the Web Service Modeling Language (WSML) based on WSMO for semantically describing ontologies, goals, and web services.

The idea of using goals, as, e.g., in WSMO, is appealing in that goals allow us to express the desired outcome of a service, i.e., as a description of the resulting system state. Note that goals thus can be formulated without reference to APIs of specific services, where merely a desired behavior can be expressed. Following Kavakli et al. [8], we say informally that a goal is a "*...a desired condition potentially attained at the end of an action (or a process)*".

This paper is a continuation of our previous work [9][10] where we presented an approach for facilitating the setting-up and management of new multi-organizational services using goals. We have in this paper also used ideas and concepts from the WSMO conceptual framework. In contrast to WSMO and OWLS-S, we propose a bottom-up approach where semantic annotations are added to the operations of each web service along the lines suggested by SAWSDL [11], and similarly to [12]. An operation annotation is a description of input/output variables, assumptions, and effects. The assumptions and effects are also expressed as goals.

We define a goal formally as a logical expression in the WSML language [13]. Note that the notion of a goal in WSML is more complex. Our aim is to let the discovery mechanism search for combinations of services that can fulfill a user-provided goal. For that purpose, we have developed a goal-refinement algorithm and incorporated it into our service discovery algorithm. Goal refinement breaks down a goal into more specific goals with which we can obtain improved matching results. Goal refinement also determines the set of operations needed for fulfilling the goal.

The contribution of this work is the combination of: (1) a goal-driven discovery mechanism for finding services, and (2) a refinement algorithm that decomposes a goal to subgoals and test whether a goal is fulfilled, and (3) a bottom up approach for semantic web service modeling with annotation at operation level. These three components make a powerful combination by extending discovery of web services by also telling exactly which operations of the discovered services that should be used in a composition.

This report is structured as follows. In Section II, we present related work. Section III, describes WSML. Section IV presents the goal-driven automated service composition framework including semantic modeling of web services. Section V explains the refinement process and provides an example. The use of refinement in the service discovery process is explained in Section VI. Section VII concludes with a summary and a discussion of the current limitations of our approach.

II. Related Work

The Universal Description, Discovery, and Integration protocol (UDDI) [14] is an industrial initiative for enabling the

ISBN: 978-1-61208-152-6

discovery and publishing of web services in a distributed way. The keyword based discovery mechanism in UDDI does not support discovery based on semantic descriptions of a service. It is thus not possible to locate a service based on what problem it solves. As a complement to UDDI, semantic matching between web services was proposed by Paolucci et al. [15]. The authors describes a model for creating semantic service profiles in DAML-S.

The Web Service Modeling Ontology (WSMO) [5] is a conceptual framework for modeling discovery and composition of web services. In WSMO, web services and goals are modeled using semantic information in the form of preconditions, postconditions, assumptions, and effects. The interaction protocol of services are described by choreographies. For service discovery, goals are matched against web service descriptions. A similar top-down approach is OWL-S [6] where a service is modeled using four ontologies: an upper Service ontology that links a *Service Profile* ontology, used for service discovery, a *Service Model* ontology, used for modeling client-service interaction and lastly a *Service Grounding* ontology, used for binding the service description into corresponding WSDL entities. A difference between OWLS-S and WSMO is that WSMO makes a distinction between goals and web services while OWL-S uses the service profile for both.

The approach of modeling services and then grounding them in WSDL files, can result in fairly large service descriptions. WSMO-lite [12] addresses this limitations of WSMO by providing a bottom-up approach where operations of a WSDL file are semantically annotated using SAWSDL [11]. This makes the service models smaller in that the service choreography automatically can be inferred from the annotations. In [16], the authors extends SAWSDL with the definition of pre- and postconditions using SPARQL query language and OWL-S Schema. The SPARQL query language is also used to formulate goals by users.

Another approach is presented in [17] where the authors introduce a goal template to precompute potential web services and goal instances that correspond to concrete client requests. Both goals and web services are modeled semantically using first-order logic as state transitions from an initial state. A similar work was presented in [18] but without goal templates. Goals are considered only as final states, as in this paper. Neither of these two papers consider annotations on operations.

The goal-based approach presented by Bandara et al. [19] uses goal elaboration based on the KAOS method [20] combined with abductive reasoning to infer the mechanisms by which a given system can achieve a particular goal. This provides for partial automation of the, possibly inconsistent and incomplete, manual KAOS approach [20]. The Goal-Based Service Framework for dynamic service discovery and composition is described in [21]. The framework allows modeling and inference between high-level goals and other parts of the system, such as a domain model, services, clients, and high-level tasks.

Table I
MESSAGING SERVICE ONTOLOGY

```
concept sendEntity
addresses ofType (1 *) _IRI
sender ofType (0 1) _string
messageIdentifier ofType _string
deliveryResult ofType (0 *) DeliveryInformation
dateTime ofType (0 1) _dateTime

concept SMSEntity subConceptOf sendEntity
message ofType _string

concept MMessageEntity subConceptOf sendEntity
subject ofType (0 1) _string
priority ofType (0 1) MessagePriority
```

III. THE WEB SERVICE MODELING LANGUAGE

WSML is defined with syntax and semantics closely following WSMO [13]. The two basic components of the WSML syntax are the *conceptual syntax* and *logical expression syntax*. The conceptual syntax is used for modeling Ontologies, Web services, Goals, and Mediators, while the logical expression syntax is used for making logical inference over these definitions. In our work, we use only a subset of the syntax: ontologies and logical expressions.

An ontology in WSML consists of concepts, their attributes and the relations between instances of the concepts. An example of a ontology for describing Messaging services is shown in Table I. A generic Message concept is encapsulated in the *sendEntity* item with the attributes, cardinality, and their value type. A more concrete instance of this messaging concept is the *SMSEntity* and the *MMessageEntity* entities, both of which inherit the attributes of their superconcept in addition to specific attributes.

The vocabulary of the logical expression syntax consists of *identifiers*, *object constructors*, *function symbols*, *datatype wrappers*, *data values* that includes all string, integer and decimal values, *anonymous identifiers*, *relation identifiers* and *variable identifiers* of the form ?alphanum*. In the proposed work, we allow a subset of WSML's logical connectives: *and*, *implies*, *impliedBy*, *equivalent*, and the auxiliary symbols: '(', ')', '[', ']', ',', '=', '!=', *memberOf*, *hasValue*, *subConceptOf*, *ofType*, and *impliesType*. For instance, "$?msg$ **memberOf** $MMessageEntity$ **implies** $?msg$ **memberOf** $sendEntity$" (where $?msg$ is a variable identifier) is a valid logical expression in the Ontology in Table I. It says that an instance of the concept *MMessageEntity* is also an instance of the concept *sendEntity*.

IV. GOAL-DRIVEN AUTOMATED SERVICE COMPOSITION

The long term vision of this work is to provide a framework where new services can easily be created. A user should be able to specify high-level business goals, in terms of QoS parameters, Key Performance Indicators and, of course, specific functionality. Then, the framework would automatically refine it into a running, self-managing service. In this context, we define a goal g to be a WSML-expression describing a desired state of a system. Thus the framework should create a service that will, when run, achieve the goal.

We have implemented a prototype of the framework consisting of three components: *service discovery engine*, *service creation engine*, and *service execution component*. Given a high-level goal, the service discovery engine tries to find a set of web service operations that fulfills the goal. The service creation engine uses the service discovery engine recursively to find all required dependencies and then composes the operations into an executable service. The service execution component then runs the composite service. In this paper, we present details of the service discovery implementation.

A. Semantic Modeling

The framework uses a bottom-up approach to semantic modeling where we annotate WSDL files with semantic information per operation. Thus we represent a web service as a set of semantically annotated operations. Each operation of a web service is modeled as a set of *input variables* I, a set of *output variables* O and a set of *state transitions* T. A transition consists of an assumption a and an effect e, where a and e are WSML-expressions. The assumption describes the state in which the operation can be applied and the effect describes the resulting state from applying the operation.

By modeling on the operation level, we can automatically generate the interaction protocol from the annotations and thus we do not need the choreography of WSMO or the Service Model of OWL-S. In addition, a goal can be much smaller than in WSMO since there is no correspondence between a goal and a web service. Instead, goals are matched to operation effects, and the assumptions can then in turn be seen as additional goals that infer dependencies between operations.

B. Semantic Matching

In line with previous work in semantic matching of web services, we define a set of categories of matching between a goal g and an operation effect e: *plugin*, *subsume*, *exact* and *no match* [15][22][7]. We use $match(g,\ e)$ to denote the type of matching, such that $match(g,\ e) = Subsume$, $match(g,\ e) = Plugin$, $match(g,\ e) = Exact$ and $match(g,\ e) = NoMatch$ denotes the corresponding matches. In a subsume match, the g is more generic than e. For instance, g subsumes e whenever $g = ?user$ **memberOf** $User$ while $e = ?user$ **memberOf** $RootUser$ given that $RootUser$ **subConceptOf** $User$. In case of a plugin match, the situation is the opposite; the e is more generic than g. For instance, if $g = ?user$ **memberOf** $User$ **and** $?user$ **memberOf** $Police$ while $e = ?user$ **memberOf** $User$, then e subsumes g and thus, it is a plugin match. In case that there are both a plugin match and a subsume match for two expressions, it is an exact match. If there is neither subsume nor plugin matches, then it is said to be a *no match.*

For implementing the matching operator, we use the algorithm implemented for checking query containment in the WSML2Reasoner framework [23]. As inference engine, we use the "IRIS well grounded" engine.

V. Goal Refinement

The discovery process starts off by the specification of a goal for a new service and proceeds by searching for sets of operations that can implement the new service by matching the goal with the effect of the operations in the annotated web services. This process is implemented using our *goal refinement algorithm*. The rest of the paper describes the implementation of goal refinement and the incorporation of it into the service discovery process.

Goal refinement requires domain knowledge of services and their components. This is described semantically by using one or more ontologies written in WSML. Algorithm 1 describes the refinement process of a goal and outputs a set of goals where each item is a refined expression of the original.

Given an input goal, the refinement algorithm decomposes the input expression into a set of atomic expressions before performing refinement on each atomic goal. Goal Decomposition, the first step of goal refinement, is the simplification of complex goals by the removal of logical connectives. This is performed using the Lloyd Topor Normalizer in the WSML2Reasoner framework [23].

Refinement of each atomic element is performed using the process described in Algorithm 1. In each atomic expression, the concept is identified (3.2) and its list of subconcepts is retrieved from knowledge represented by the ontology (3.3).

Algorithm 1 Simple Goal Refinement

```
procedure RefineSimpleGoal($g$ - a goal)
    $G' \leftarrow decompose(g)$                        ▷ - {3.1}
    $SG \leftarrow \emptyset$                               ▷ - a set of subgoals
    for all $g' \in G'$ do
        $c \leftarrow findConcept(g')$                   ▷ - {3.2}
        $SC \leftarrow listSubConcepts(c)$               ▷ - {3.3}
        if $SC$ is not empty then
            for all $sc \in SC$ do
                $g'' \leftarrow g'$
                replace $c$ in $g''$ with $sc$
                insert $g''$ into $SG$
            end for
        end if
    end for
    return $SG$
end procedure
```

We illustrate the algorithm by refining a simple goal. The ontology in Table I provides the context for the goal, and a messaging concept is represented using an entity *sendEntity* which consists of two smaller entities, *SMSEntity* and *MMSEntity*. The former represents a simple text message and the latter a multimedia message. Assume that we are given a goal requesting a generic messaging service. This could be the case when the requester doesn't know or doesn't care how a message should be forwarded. Since the goal is generic we may assume that there is no service available matching the

Table II
INPUT AND REFINED OUTPUT EXPRESSIONS

Input Expression:
?x memberOf msgOnt#sendEntity.

Output Expression Set:
?x memberOf msgOnt#MMSEntity.
?x memberOf msgOnt#SMSEntity.

goal, and that we thus need to refine it into more specific goals which in turn may be matched by existing services.

The goal expresses a requirement for a generic messaging service, described as the input expression in Table II. The generic messaging ontology given above, with its two subconcepts is used for refining the goal. The goal is refined into two subgoals containing the two subconcepts. Table II shows the original (input) and refined (output) expressions from the algorithm. Each subgoal may be satisfied by different service providers: a text messaging service provider and a multimedia messaging service provider. Provision of both services will fulfill each of the subgoals and in turn the original goal.

This example illustrated that a single provider may not possess all the resources necessary to fulfil a goal and that refinement into more specific subgoals may help aggregating providers whose services may fully match the given goal.

In the discovery process (see Section VI), refinement is performed when a plugin or subsume match has been obtained during the matching of a goal and a web service operation. In the case of a plugin match between a service provider and a goal consumer, the goal is more specific than the web service. A refinement of the web service is then performed and each refined web service is matched with the goal. With a subsume match, the web service is more specific than the goal, in which case the goal is refined and for each refined subgoal a matching with the web service is performed.

VI. GOAL REFINEMENT IN SERVICE DISCOVERY

Service discovery is achieved by matching a goal provided by a user against the operations of all web services using goal refinement. By integrating goal refinement into the discovery process, we improve match results by being able to combine subsume and plugin type matches. In addition, we extract common and differing parts of expressions when matching a goal to a web service operation. This information is used to modify the goal if necessary and in further matching processes.

We produce three sets of expressions in this extraction process. The common expressions, *CO*, represent the resources required by the consumer to be fulfilled by the provider. Goals required by consumer that cannot be fulfilled by the provider, *GO*, form a new goal and is used to match against services by other providers. Providers who offer resources that are not needed by the consumer, *WO*, can be offered to other consumers, thus optimizing resource usage. We demonstrate in Algorithm 2 the process of comparing a goal and a web service with the refinement and extraction processes introduced.

By comparing all the operations in a web service against a goal, as implemented in Algorithm 2, we select the operation that best fulfills a goal. This is done by selecting the operation that has the most in common with the goal, while keeping the unfulfilled goal size to a minimum and using as much of the web service operation as possible.

For each web service operation and goal, we use Algorithm 2 to obtain the matching result, rank these results and select the most suitable web service operation. The goals left unfulfilled by the selected web service operation are formulated as a new goal which is to be matched against other web services in the next iteration of the refinement. In the current version of our implementation, only one operation is selected from each matching web service.

Algorithm 2 Refinement in Matching

```
procedure ($g$ - a goal, $e$ - an effect)
    $C \leftarrow \emptyset$                         ▷ : In both goal and effect.
    $GO \leftarrow \emptyset$                        ▷ :Found only in goal.
    $WO \leftarrow \emptyset$                        ▷ :Found only in effect.
    $G' \leftarrow$ decompose($g$); $E' \leftarrow$ decompose($e$)
    for all $g' \in G'$ do
        for all $e' \in E'$ do
            $type \leftarrow match(g', e')$
            if $type = Exact$ then
                insert $g'$ into $C$; remove $e'$ from $E'$
                matched $\leftarrow$ true
            else if $type = Subsume$ then
                Handle subsume               ▷ :See Algorithm 3.
                remove $e'$ from $E'$; matched $\leftarrow$ true
            else if $type = Plugin$ then
                Handle Plugin                ▷ :See Algorithm 4.
                remove $e'$ from $E'$; matched $\leftarrow$ true
            end if
            if $matched = true$ then
                break loop
            end if
        end for
        if $matched = false$ then
            insert $g'$ into $GO$
        end if
    end for
    $WO \leftarrow E'$
    return $\langle C, GO, WO \rangle$
end procedure
```

Algorithm 3 Handle Subsume Matches

```
$SG \leftarrow refine(g)$                ▷ : Refine Goal into Subgoals.
for all $sg \in SG$ do
    $subtype \leftarrow match(sg, e)$
    if $subtype = NoMatch$ then
        insert $sg$ into $GO$
    else if $subtype = Exact$ then
        insert $sg$ into $C$
    else
        Compare attributes between sg and e
        Return common, onlyGoal and onlyWS attributes
    end if
end for
```

Algorithm 4 Handle Plugin Matches

```
$E' \leftarrow refine(e)$                ▷ : Refine web service effects.
for all $e' \in E'$ do
    $subtype \leftarrow match(g, e')$
    if $subtype = NoMatch$ then
        insert $e'$ into $WO$
    else if $subtype = Exact$ then
        insert $e'$ into $C$
    else
        Compare attributes between g and e'
        Return common, onlyGoal and onlyWS attributes
    end if
end for
```

A. An Example

We provide an example on service discovery performed together with refinement and extraction methods. Our user-provided goal is to create a service that is able to deliver a message to a registered user. A number of ontologies are used to describe this goal and a number of web service providers have services available for the automated composition process. In our tests, we have modeled an Information service, SMS service and MMS service among others. Table III shows the user-provided goal, $G0$ and some operations ($OP1$, $OP2$, and $OP3$) in available web services. Table IV shows the results from attempting to match $G0$ with each of the web services.

As implemented in Algorithm 2, the goal is first decomposed into a set of *atomic* (with respect to the refinement algorithm) goals and each component is compared against the web service operations. The original goal expression, $G0$ is compared to the first service (information service) and obtains the best match against the operation *createUser (OP1)*. The *createUser* web service operation partially fulfills the goal. The results obtained with the matching of each atomic expression are either exact matches or no match. In this step, no plugin or subsume matches have been obtained to warrant refinement. *F1* of the goal is fulfilled while the remaining unfulfilled goal is formulated into a new goal $G1$ and used in subsequent matches with other web services.

$G1$ is matched with the operations in the next web service, an SMS web service. *F2* shows the goals that can be fulfilled by the *sendSMS (OP2)* operation in the web service and the remaining unfulfillable goals $G2$ are used in subsequent matches. In this comparison, the atomic expression in $G1$, *?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#sendEntity*, is a subsume match to the atomic expression of *?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#SMSEntity*. Refinement of the $G1$ yields two subgoals: *?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#SMSEntity* and *?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#MMSEntity*. The former is an exact match and the latter, $G2$, is added to the list of goals that need to be fulfilled by other providers.

$G2$ is matched against the operations in the next web service, an MMS web service. *F3* shows the goals that can be fulfilled by the *sendMMS* operation in the web service and the remaining *G3* remains unfulfilled.

Table III
COMPARING GOAL AND WEB SERVICE EXPRESSIONS

Goal Expression, *G0* : ?user[ug#name hasValue "Tomas Olsson", ug#mobilePhone hasValue ?mp] memberOf ug#User and my#isRegisteredByInfoService(?user) and ?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#sendEntity and msgOnt#forDelivery(?msg).
Create User Operation from Information Service, *0P1*: ?newUser[ug#name hasValue ?userName] memberOf ug#User and my#isRegisteredByInfoService(?newUser). **Send SMS Operation from SMS Service, *0P2*:** ?x[msgOnt#address hasValue ?address] memberOf msgOnt#DeliveryInformation and ?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#SMSEntity and msgOnt#forDelivery(?msg). **Send MMS Operation from MMS Service, *0P3*:** ?x[msgOnt#address hasValue ?address] memberOf msgOnt#DeliveryInformation and ?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#MMSEntity and msgOnt#forDelivery(?msg).

VII. CONCLUSION AND FUTURE WORK

In this paper, we have described a goal-driven service discovery mechanism and an approach to annotating operations in semantic web services. We have described how our goal refinement algorithm can be used to refine a user provided goal into subgoals with which we can perform a fine-grained service matching in the service discovery process. Moreover, we have illustrated how non-matched subgoals can be extracted to form new goals for subsequent matching.

Our discovery process has not been optimised to reduce the time required to search for the best matching web service to the goal. Currently, the larger the number of web services in the repository, the more time is required to complete a matching. The complexity of the algorithms needs further investigation, and the algorithms should be improved accordingly. In our implementation, the order in which web services are matched affects the results. The resulting composition may be improved (e.g., with respect to the number of implementing services) by investigating methods for finding optimal orderings of the services prior to matching.

Using goals for modeling and finding services seems to be a promising approach. By specifying high-level goals it is possible to manage services at macro level and derive management of specific services at micro level using goal-refinement. In this work, we have used implicit goal-decomposition and goal-composition rules. However, to make the approach more flexible we need an explicit goal-translation modeling language, where a domain expert can explicitly model what high-level goals can be translated into what low-level goals.

Table IV
SERVICE DISCOVERY RESULTS

Results of Comparison Against Information Service Goal Fulfilled, *F1*: ?user[ug#name hasValue ”Tomas Olsson”] memberOf ug#User and my#isRegisteredByInfoService(?user). Remaining Goal, *G1*: ?user[ug#mobilePhone hasValue ?mp] memberOf ug#User and ?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#sendEntity and msgOnt#forDelivery(?msg). Best Operation Match: http://www.sics.se/gops/webservices/InfoService/createUser
Results of Comparison Against SMS Service Goal Fulfilled, *F2*: ?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#SMSEntity and msgOnt#forDelivery(?msg). Remaining Goal, *G2*: ?user[ug#mobilePhone” hasValue ?mp] memberOf ug#User and ?msg[msgOnt#deliveryResult hasValue ?x] memberOf msgOnt#MMSEntity. Best Operation Match: http://www.sics.se/gops/webservices/SMSService/sendSMS
Results of Comparison Against MMS Service Goal Fulfilled, *F3*: ?msg[msgOnt#deliveryResult” hasValue ?x] memberOf msgOnt#MMSEntity”. Remaining Goal, *G3*: ?user[ug#mobilePhone hasValue ?mp] memberOf ug#User. Best Operation Match: http://www.sics.se/gops/webservices/MMSService/sendMMS
Final Results Final Set of Operations: http://www.sics.se/gops/webservices/InfoService/createUser http://www.sics.se/gops/webservices/SMSService/sendSMS http://www.sics.se/gops/webservices/MMSService/sendMMS Goal not fulfillable by any service: ?user[ug#mobilePhone hasValue ?mp] memberOf ug#User.

ACKNOWLEDGEMENT

The present work was funded by the GOPS project. GOPS is sponsored by Vinnova.

REFERENCES

[1] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web services description language (wsdl) 1.1,” World Wide Web Consortium, W3C Note 15, March 2001.

[2] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon, “Soap version 1.2 part 1: Messaging framework (second edition),” World Wide Web Consortium, W3C Recommendation 27, April 2007.

[3] J. Garofalakis, Y. Panagis, E. Sakkopoulos, and A. Tsakalidis, “Web service discovery mechanisms: Looking for a needle in a haystack?” in *International Workshop on Web Engineering*, August 2004.

[4] N. Steinmetz, H. Lausen, and M. Brunner, “Web service search on large scale,” in *Proceedings of the 7th International Joint Conference on Service-Oriented Computing*, ser. ICSOC-ServiceWave ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 437–444.

[5] D. Roman and H. Lausen, “D2v1.3. web service modeling ontology (wsmo) - wsmo final draft 21 october 2006,” [Online]. Available: http://www.wsmo.org/TR/d2/v1.3/ 29-06-2011.

[6] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, “Owl-s: Semantic markup for web services,” [Online]. Available: http://www.ai.sri.com/daml/services/owl-s/1.2/overview/ 29-06-2011.

[7] U. Keller, R. Lara, H. Lausen, A. Polleres, L. Predoiu, and I. Toma, “D10v0.2 semantic web service discovery wsmx working draft,” [Online]. Available: http://www.wsmo.org/TR/d10/v0.2/ 29-06-2011.

[8] E. Kavakli and P. Loucopoulos, *Information Modelling Methods and Methodologies*, 2005, ch. Goal Modelling in Requirements Engineering: Analysis and Critique of Current Methods, pp. 102–124.

[9] Å. Berglund, B. Bjurling, R. Dantas, S. Engberg, P. Giambiagi, and B. Ohlman, “Towards goal-based autonomic networking,” in *Third International Workshop on Distributed Autonomous Network Management Systems*, Nov 2008.

[10] M. Y. Chong, B. Bjurling, R. Dantas, C. Kamienski, and B. Ohlman, “Goal-based service creation using autonomic entities,” in *MACE ’09: Proceedings of the 4th IEEE International Workshop on Modelling Autonomic Communications Environments*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 29–43.

[11] J. Farrell and H. Lausen, “Semantic annotations for wsdl and xml schema,” World Wide Web Consortium, W3C Recommendation 28, August 2007. [Online]. Available: http://www.w3.org/TR/sawsdl/

[12] T. Vitvar, J. Kopecký, J. Viskova, and D. Fensel, “Wsmo-lite annotations for web services,” in *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ser. ESWC’08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 674–689.

[13] N. Steinmetz and I. Toma, “D16.1v1.0 wsml language reference - wsml final draft 2008-08-08,” [Online]. Available: http://www.wsmo.org/TR/d16/d16.1/v1.0/ 29-06-2011.

[14] “Uddi,” http://uddi.xml.org/ 29-06-2011.

[15] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, “Semantic Matching of Web Services Capabilities,” in *The Semantic Web - ISWC 2002: First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002. Proceedings*, 2002, pp. 333+.

[16] K. Iqbal, M. L. Sbodio, V. Peristeras, and G. Giuliani, “Semantic service discovery using sawsdl and sparql,” in *Proceedings of the 2008 Fourth International Conference on Semantics, Knowledge and Grid*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 205–212.

[17] M. Stollberg, U. Keller, H. Lausen, and S. Heymans, “Two-phase web service discovery based on rich functional descriptions,” in *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, ser. ESWC ’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 99–113.

[18] R. Lara, “Two-phased web service discovery,” in *AI-Driven Technologies for Services-Oriented Computing Workshop at AAAI-06*, Boston, USA, 2006.

[19] A. K. Bandara, E. Lupu, J. D. Moffett, and A. Russo, “A Goal-based Approach to Policy Refinement,” in 5th *IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY’04)*, 2004, pp. 229–239.

[20] R. Darimont and A. van Lamsweerde, “Formal Refinement Patterns for Goal-Driven Requirements Elaboration,” in *4th ACM SIGSOFT Symposium on Foundations of Software Engineering*, vol. 21, no. 6. ACM Press, Nov 1996, pp. 179–190.

[21] L. O. B. da Silva Santos, G. Guizzardi, L. F. Pires, and M. van Sinderen, “From user goals to service discovery and composition,” in *Advances in Conceptual Modeling - Challenging Perspectives*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, November 2009, vol. 5833, pp. 265–274.

[22] U. Keller, R. Lara, A. Polleres, I. Toma, M. Kifer, and D. Fensel, “WSMO Web Service Discovery,” DERI, Tech. Rep. D5.1v0.1, November 2004.

[23] S. Grimm, U. Keller, H. Lausen, and G. Nágypál, “A reasoning framework for rule-based wsml,” in *In Proceedings of 4th European Semantic Web Conference (ESWC)*, 2007.

Automatic Service Retrieval in Converged Environments Based on Natural Language Request

Edgar Camilo Pedraza
GIT
University of Cauca
Popayán, Colombia
epedraza@unicauca.edu.co

Julián Andrés Zúñiga
GIT
University of Cauca
Popayán, Colombia
gzja@unicauca.edu.co

Luis Javier Suarez Meza
GIT
University of Cauca
Popayán, Colombia
ljsuarez@unicauca.edu.co

Juan Carlos Corrales
GIT
University of Cauca
Popayán, Colombia
jcorral@unicauca.edu.co

***Abstract*—Finding services in dynamic and heterogeneous contexts, as converged environments (Next Generation Networks), is a very complex task and a crucial aspect in this new paradigm of convergence. Therefore, it is essential to have efficient and effective mechanisms for seeking services, to take advantage of resources in the network (Web and telecommunications services). Recent studies have developed Service Creation Environments for Telecommunications and Internet converged services, where user´s requests are represented by complex expressions that describe the required services. Thus, the search and selection of these services depend on the ability of the developer to retrieve the most suitable ones, converting this labor in an inefficient work. With this in mind, and in order to improve the time to create convergent services, this paper proposes a novel approach that supports the automatic retrieval of services in converged environments, considering functional and non-functional requirements of end-user's requests in natural language, to optimize the process of convergent services creation.**

***Keywords-automatic service retrieval; converged environments; natural language request; Telecom and Internet converged service*.**

I. INTRODUCTION

The ability to retrieve services that accomplish user requests, has led to the development of various projects, focused on service retrieval, understood as an important stage in the composition process [1] that allows the user to find and use a service based on a published description of its functionality or operational parameters [2]. Currently, services retrieval offers new challenges driven by the convergence of Web and telecommunications domains around the IP protocol, enabling the use of diverse and innovative services, regardless of the customer access network [3]. The above-mentioned, both with new trends in application environments, where users are important generators of content and applications, opens up towards a new paradigm in which non-technical individuals are able to design and create their own fully customized services by integrating Web and telecommunication components, an activity that years ago was done only by expert developers due to its complexity.

From the above notion, in this paper, we propose an architecture for automatic service retrieval in converged environments, considering the services functional properties (e.g., inputs, outputs, preconditions and effects) and nonfunctional properties (e.g., QoS, such as: availability, response time, reputation, etc) requested by the user in natural language (NL), to speed up the creation of converged services.

Typically, natural language processing (NLP) is useful to analyze and produce semantic representations of the user's request, providing information needed to identify the generic control flow, as well as functional and nonfunctional properties of services. Therefore, in this paper, we propose the use of NLP techniques to automate the process of service retrieval from requests made in NL. Thus, with semantic descriptions supported by NLP techniques and adaptation of algorithms for matching services and user's request, it is possible to make an accurate and automatic retrieval of services available within both Web and telecommunications domains.

The remainder of this paper is structured as follows: in the next section, we review the work related to the different topics involving the current research. Then, in Section III, a high level description of the proposed architecture is presented. To provide greater clarity an example is discussed in Section IV. Finally, in Section V we conclude the paper.

II. RELATED WORK

Service retrieval can be addressed under two main approaches: syntactic and semantic. The searching of services from a syntactic approach, considers either, interfaces matching techniques or keyword searching [4] that require exact matches at the syntactic level between the descriptions of services and the parameters used, which leads to deficient results in the retrieval of service. The semantic approach allows the establishment of relationships between concepts that define the functionality of services (functional properties) and additionally, considers formal descriptions constructed by non-functional properties [5], achieving a more precise description of services, improving the quality of results to retrieve services according to user needs [6].

From the foregoing, and given the nature of the problem, the proposed solution focuses on services retrieval based on semantics and NLP. In this sense, some existing solutions are described below. In [7], the IBM research team developed a supercomputer that performs analysis phases of natural language questions composed by hundreds of algorithms, some of these phases present a similar approach with the

ISBN: 978-1-61208-152-6

proposed ones in this paper. However, it requires a complex system composed of multi-core hardware processor.

In [8], the authors address the selection of Web services based on requests expressed in NL, this solution is based on language restrictions, matching the structure of the request with predefined patterns for decomposition into blocks using keywords. Subsequently, the request is processed and transformed into a data flow and control model expressing the general logic of the new service. In addition, the authors propose the use of a common ontology and a NL dictionary, in order to relate textual fragments with functional parameters of such services. This paper presents disadvantages when limiting requests to simple sentences.

Based on concepts graphs and conceptual distance measure, a solution is presented in [1]. Its purpose is to calculate the similarity between the user's request, represented by keywords, and services available in a repository. Within the linguistic analysis, different processes are performed: text segmentation, irrelevant word removal, elimination of derivatives (stemming) and grammatical corrections. The authors admit their proposal's lack of dynamic adaptation at runtime.

The approach of [5] re-uses the converged services creation environment of project SPICE (Service Platform for Innovative Communication Environment) [9, 10] to facilitate services retrieval with different types of semantic annotations. The author focuses his work on the development of an intelligent agent in charge of analyzing the application in NL to extract semantic information, specifically the goals, from which additional semantic information is derived, as inputs and outputs, which are used to retrieve services and report their order composition. However, retrieval's throughput and processing critically depend on the amount of services stored in the repository.

From the previous review, limitations are evident because they are based on functional preferences, leaving aside non-functional requirements that provide great sense of services semantic descriptions. Finally, the automatic retrieval of services in converged environments is a recent topic of research, where, considering the above characteristics, no work has been done.

III. ARCHITECTURE

In this section, we present our proposed architecture for automatic services retrieval in converged environments, which receives as input the user's request made in NL from a mobile device, and gives a service ranking and a generic control flow, as output. Figure 1 shows the modules of the architecture, organized in four phases, which correspond to: *Natural Language Analysis, Matching, Recommender and Inference phase*, these modules are described below:

- *Tokenizer*: as input, it has the NL request and from this, it obtains words, phrases or symbols called tokens.
- *Filter Words*: responsible for removing non-sense words by comparing with a set of words previously identified.
- *Words Tagging*: tags words according to its grammatical category (e.g., she "pronoun”, loves "verb", animals "noun").
- *Named Entity Recognition*: classifies the words into “functional” or “control” categories.
- *Semantic Analyzer*: in charge of semantic disambiguation process of input words.
- *Non Functional Requirements Recommender:* searches non-functional parameters in the repository from functional request previously written by user.
- *Services Recommender:* searches request-service information in the repository obtained from prior inputs of users.
- *Matcher Functional Requirements*: obtains from cluster services, the first rank, by matching functional requirements.
- *Ranking Generator*: obtains the final ranking of services considering, if exists, non-functional requirements, of services, such as QoS.
- *Cluster of Services*: conformed by abstract descriptions of Web and Telecommunications domain services, which are conceptually organized as functional properties.
- *Upper Ontology:* involves general concepts that are the same across all knowledge domains (e.g. QoS, Telco, IT, among many others), supporting the functional and non-functional properties description and enabling the ontology reasoning.
- *Flow Ranking Repository:* stores an association of service (tags) obtained at the end of the semantic matching phase.
- *Non functional Repository:* stores an association of non-functional and functional parameters of cluster´s services.
- *Generic Flow Generator*: generates an approximate generic control flow, based on keywords taken from the user request processed.
- *Flow-Ranking Associator:* associates the flow obtained in the *Generic Flow Generator* module with the ranking output of the matching semantic phase, obtaining the final output of the architecture.

Most relationships between architecture modules are sequential. However, there are interactions between stages that do not follow this behavior. Below, a more detailed description is made, of the different phases and processes that take place inside of them.

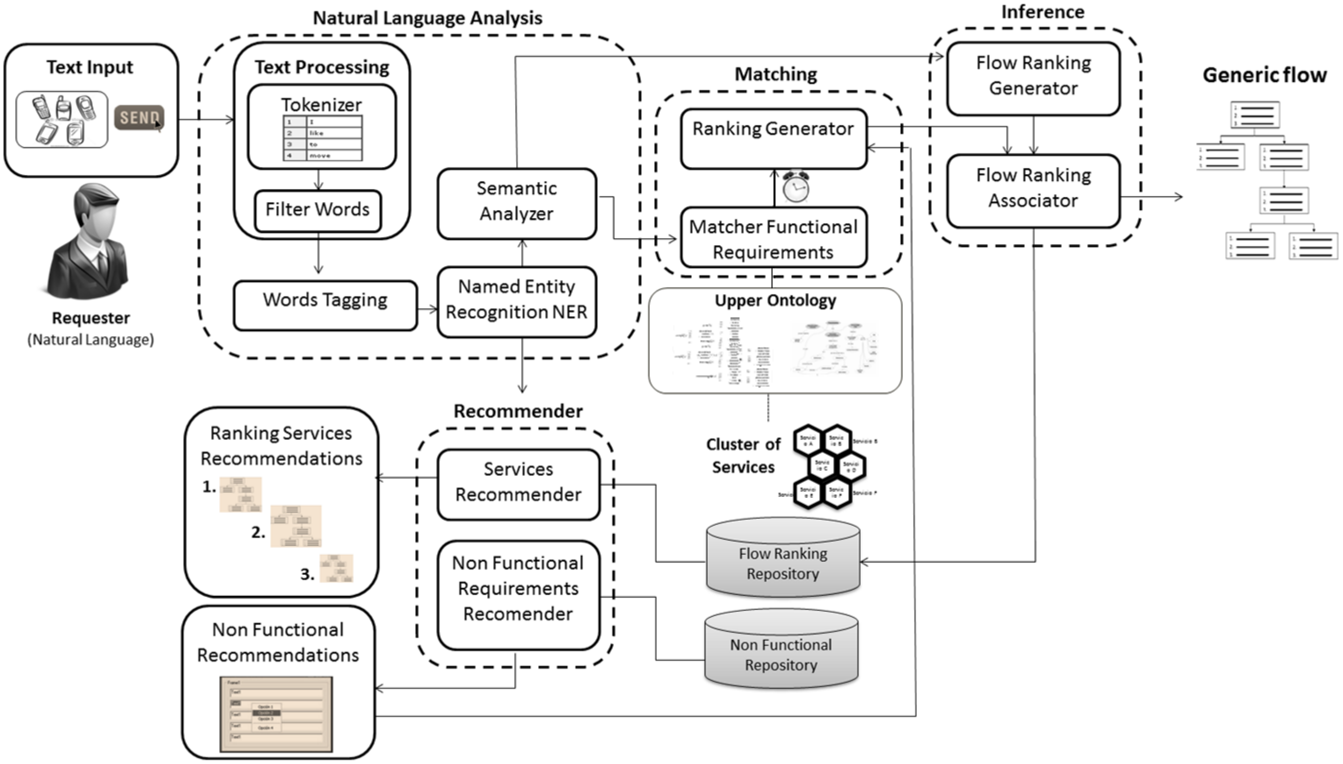


Figure 1. Architecture of the System.

A. Phase of Natural Language Analysis (NLA)

Initially, a user makes a request from his/her mobile device in NL, which is received by the *Tokenizer*, where tokenization operation starts, it obtains simple lexical units from complex sentences, by removing existing spaces. Additionally, this module corrects simple lexical errors that may arise in the request, i.e., misspelled word errors that are easily identifiable. Afterward, the sentences are processed through *Filter Words Module* and later they pass through *Words Tagging*, with which, it is pretended to classify (tag) words of the sentence according to their grammatical category. The module also aims to undertake an analysis based on linguistic rules, trying to identify and compensate syntactic and structural errors. Some techniques used to implement the modules above are GateNLP, OpenNLP, Apache UIMA, among others. One of the most important is GateNLP [12], which offers an architecture that contains functionality for plugging in all kinds of NLP software: (POS taggers, sentence splitters, named entity recognizers) and all are java based.

Once completed these operations, the request is more consistent, but remains complex. Therefore, we established the *Named Entity Recognition*, which performs a classification between "*Control"* and *"Functional"* words according to its meaning, from which, control words are directed to Flow Ranking Generator Module, whereas functional words are directed to *Semantic Analyzer* and also to *Recommender modules*. Within the final stage of linguistic analysis, it is important to consider the semantic ambiguity, for which the *Semantic Analyzer* identifies the correct sense of words according to their context, i.e., identifies the correct one from a word within multiple meanings that can occur in a sentence. This allows an easy identification of keywords with their respective grammatical category (e.g., noun, verb, adjective, conjunction) that define the user's request and with which service selection will be made. Thus, this stage offers the user greater flexibility in the use of language and allows the establishment of a wider range of possibilities. On the other hand, this phase also allows conditional words identification (e.g., *if*, *then*, *later*) and words of order (sequence) (e.g., *first*, *second*) important for the *Generic Flow Generator* in the inference phase.

B. Recommender

This phase is composed of two modules and it is executed while matching phase performs the searched of services with functional requirements. Additionally, the phase shows a hint to the user of generic control flow stored in a repository, if the user select one, the execution of the two remaining phases would be avoided, and straightaway the process finished, otherwise, the process continues in the matching phase. For the above task, the module in charged is the *Service Recommender*, which obtains a list of generic control flow with services of the repository from obtained keywords classified as functional. The second one module is named *Non Functional Requirements Recommender*, this searches non-functional parameters from the *Non*

Functional Repository using the obtained keywords classified as functional, the result is shown to the user, and if he/she selects one, it becomes the input of the *Ranking Generator* module.

C. Matching Phase

At this stage there are two important modules: the first one is related with the use of an upper ontology for the matching of functional requirements, to obtain the first ranking of services (*Matcher Functional Requirements*). The second one refers to the Generator Ranking module which use a *weighting algorithm*, once the first ranking of services is obtained, it weights ranking with non-functional keywords which are taken from the Recommender phase, for which, the *Generator Ranking* module uses a timer. It provides a time out for an entry of those parameters. If after a certain time, income of non-functional words does not exist, the module will get a ranking with only functional parameters. Non-functional parameters are very important in the request because they represent aspects such as quality, efficiency, availability, etc., with which is possible to offer more adequate services to users. It considers that request can be enriched with non-functional parameters, in order to provide optimum results that best fit to end user requirements.

D. Inference phase

This stage begins with the *Generic Flow Generator*, which receives as input Keywords classified as Control obtained from NER through NLA phase. With this, the module infers a basic structure of ordered operations that represents the basic control flow, useful for the composition of services, which is performed after the service retrieval. *Flow-Ranking Associator* receives as inputs a service ranking from the semantic matching phase and the basic control flow (obtained from the previous module) in order to generate the services generic control flow that is stored in *Flow Ranking Repository* and becomes the output of the whole architectures.

IV. Example

This section describes the functionality of the proposed architecture through an example that details each of the phases of the process for automatic retrieval of services in converged environments. To do so, consider the following situation. Using natural language, an executive requests from his cell phone a meetings coordination service, so: *"I want to receive traffic reports of Bogotá via messages, minutes before the meeting and if I have not made it to the meeting, I want to receive audio content of the decisions taken."*

A. Information Retrieval with Natural Language Analysis

1) Tokenizer: the result of this procedure is as follows:

"*I –Want –to- receive – traffic – reports – of – Bogotá…*"

2) Filter Words: removes unimportant words for the request (in this case, words as: I - want are removed), resulting*:*

"To r*eceive – traffic – reports – Bogotá…*"

3) Words Tagging: labels and classifies the words obtained before as follows*:*

"To *receive: Verb – traffic: Noun...*"

4) Named Entity Recognition: the system classifies the words into "functional" or "control" categories, grouping the words of the functional category in blocks separated by words of the control category, resulting in:

"Block 1: Receive (functional) –traffic(functional) – reports(functional)… Control: before (control) – and(control) – if(control)…Block 2… "

5) Semantic Analyzer: at this point we detail the semantic disambiguation of the words classified as functional, based on dictionary [11]. For this example, "report" can be verb or noun, and may have several meanings including: *a written document describing the findings of some individual or group*, *a short account of the news*, and others, from which the system determines the second choice as relevant to this case, using a variant of the *Lesk algorithm* mentioned in [9].

B. Recommender

1) Recommender Services: it compares obtained keywords classified as functional with a flow ranking recommendation repository, considering the case for the existence of some match with the words: *"traffic reports"*, the result shown to the executive, is a list of generic flows with services, outcome from previous requests to the system:

"First: SendSMS to GetTraffic …
Second: GetPosition to GetTraffic to SendSMS …"

The executive doesn't select any recommendation, so the remaining processing continues.

2) Non Functional Requirements Recommender: this recommender searches non functional parameters from the *Non Functional Repository* using the obtained keywords classified as functional, the result, considering the case for the existence of matches with the words: *"traffic reports"* is as follows:

"*Precision, real-time ...*"

The executive chooses the option *"precision"* and it's accepted by the system.

C. Semantic comparison between the processed request and Service Cluster

1) Matcher functional parameters: the input for this stage is represented by two blocks: *" receive traffic reports Bogotá message minute meeting"* and *"receive content audio decision"*, now assuming that, from the process of comparison and service retrieval, the following services were obtained: for the first block: *SendSMS, GetTraffic, GetSMS, SetUpMeeting, AlertMeeting, GetMMS*, for the second block: *GetAudioContent, SendAudioContent, SendMMS, getMMS, GetPosition*. These services are organized according to the functional parameters comparison (goals, inputs and outputs) getting two ranking of services for each input block respectively: *1-AlertMeeting, 2-GetTraffic,3-GetSMS, 4-GetMMS, 5-SetUpMeeting and 6-SendSMS; and 1-GetPosition, 2-GetMMS, 3-GetAudioContent, 4-SendAudioContent, 5-SendMMS*.

2) Ranking Generator: The system waits a determined time for the input of non-functional parameters, as the executive chose the parameter "*precision*", the retrieved ranking of service is subjected to a weighting based on this non-functional parameter, generating a new ranking for the two blocks: *1-AlertMeeting, 2-GetTraffic, 3-GetMMS, 4-GetSMS, 5-SendSMS and 6-SetUpMeeting; and 1-GetPosition 2-GetAudioContent, 3-GetMMS, 4-SendAudioContent and 5- SendMMS.*

D. Retrieval-based inference

1) Generation of the generic flow control: as a result we have two generic blocks and the words classified as control. *"before and if not"*.

2) Association of ranking and flow: in this phase, the system replaces the generic blocks generated in the previous phase, for the list of services retrieved for each block.

3) Result Storage: the words are stored keeping a relationship with the services that were retrieved, given the possibility of future requests, reduces processing time.

V. CONCLUSION

In this article, we presented an approach that speeds up creation of services in converged environments, critical issues in service deployment by companies in the telecommunications sector. The proposed architecture starts off from a request made by the user in NL from a mobile device and delivers a generic control flow as output which identifies the most suitable services to users. Our proposal also includes the use of top level ontology, which provides greater performance at services selection and also uses a requests-services repository, where records that speed up retrieval time are stored. No incoming requests restriction allows inexperienced users to make requests to the system, unlike other solutions that use templates restricting user's expression.

As a complementary work, we can consider methods for making non-textual requests, like voice, adding a linguistic level (phonetic) to the NLP, increasing the range of end-users. Also, we do not discard the possibility of considering user information, such as Profile and Context.

REFERENCES

[1] Pop F.-C., Cremene M., Tigli J.-Y., Lavirotte S., Riveill M., and Vaida M., Natural Language based On-demand Service Composition. International Journal of Computers, Communications & Control, 2010. V (4): pp. 871-883.

[2] Bandara, A., Semantic Description and Matching of Services for Pervasive Environments, in Engineering, Science and Mathematics 2008, Universidad de Southampton Southampton. pp. 100-150.

[3] ITU-T, General overview of NGN, in Scope and Propose 2004, ITU-T. pp. 10.

[4] Corrales, J.C., Behavioral matchmaking for service retrieval, in Computer Science 2008, University of Versailles Saint-Quentin-en-Yvelines: Versailles. pp. 88.

[5] Sutthikulphanich, K., A Demonstration on Service Compositions based on Natural Language Request and User Contexts, in Telematics 2008, Norwegian University of Science and Technology: Trondheim. pp. 172.

[6] Al-Masri, E. and Q. Mahmoud, Discovering the Best Web Service:A Neural Network-based Solution, in Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics 2009: San Antonio, TX, USA. pp. 4250-4255.

[7] IBM (2011). "What is Watson?" Retrieved August 13, 2011, from http://www-03.ibm.com/innovation/us/watson/what-is-watson/index.html.

[8] Bosca, A., Corno F., Valetto G., and Maglione G., On the fly Construction of Web Services Compositions from Natural Language Requests. Journal of Software, 2006. 1(1): pp. 40-50.

[9] Tarkoma, S., Prehofer, C., Zhdanova A., Moessner K., and Kovacs E., (2007). SPICE: Evolving IMS to Next Generation Service Platforms. International Symposium on Applications and the Internet Workshops. Hiroshima, Japan: pp. 6.

[10] Cordier, C., and Kranenburg, H., Specification of the Knowledge Management Framework, The SPICE project, January 2006, pp 1-49.

[11] Nica, I., Automatic semantic disambiguation, in Language and communication 2002, University of Barcelona: Barcelona.

[12] GateNLP (2011). "Developing Language Processing Components with GATE Version 6" Retrieved August 15, 2011 from http://gate.ac.uk/sale/tao/split.html.

Using Statistical Tests in a Trust Model

Francisco Javier Nieto
ATOS Research and Innovation
Atos Origin
Bilbao, Spain
Francisco.nieto@atosresearch.eu

Abstract—**Nowadays, it is widely recognized that trust is a key aspect of services when integrating them in enterprise environments. This paper presents a solution which takes advantage of the data available by carrying out statistical hypothesis tests in order to evaluate some aspects which are directly related to a service trust. As a result, the application of some tests improves the accuracy and increases the robustness of trust models.**

Keywords-trust; services; statistical tests; model

I. Introduction

As services have become the key for enabling interoperability between enterprises and systems in general, it is necessary to guarantee that interactions are performed properly. It is easy to imagine a scenario where a cluster of companies work together in order to obtain a final product (i.e., suppliers and providers interacting in the automotive domain for producing cars, where many external services are used, such as currency converters or even traffic information providers for logistics).

Services may become very important pieces integrated in business processes or in systems which require that third parties provide functionalities in a professional way or, at least, fulfilling a minimum quality levels in a secure way. For this reason, it is necessary to include mechanisms which guarantee that services provide functionalities as expected and that they deal internally with the information in a confidential and secure way.

The analysis performed in [1] already revealed that there are two kinds of mechanisms: hard security mechanisms and soft security mechanisms. In this case, this paper is focused on a soft security mechanism for calculating trust, understanding it as the belief in the reliability, truth and capability of the service. Such a solution could be used by service discovery tools, as a way to filter services to be listed as candidates.

Despite of the existence of several solutions and the availability of a lot of data about services, no formal analysis is performed about the data gathered as a mean to improve the accuracy and robustness of trust models.

Information about a service can be provided from users who invoked it, but nowadays there are other sources such as monitoring tools (gathering information each invocation, like response time or availability) and platforms collaborating in federations. The presented approach aims at exploiting all this information in a trust model by applying several statistical hypothesis tests. Depending on the data available and the test, it is possible to determine an evaluation of concrete aspects which have great importance upon the global trust of a service, so these evaluations will be aggregated later to other aspects in a higher level model in order to obtain an accurate enough measure of the trust.

The paper is structured as follows: Section 2 provides a vision of the related work in the area, while Section 3 describes the main objectives on which the approach is based. Section 4 will describe the usage of tests for checking agreements fulfillment, Section 5 presents those tests used for evaluating the successfulness of a new release and Section 6 is focused on how to apply the tests when aggregating opinions. Finally, Section 7 presents a set of conclusions and future work.

II. State of the Art

There are multiple initiatives which have analyzed the best way to aggregate users' opinions in order to determine the trust and the reputation of services and systems. Each solution depends on two factors: the kind of information to be used (a general opinion or if there are some aspects to be aggregated) and the way to calculate the aggregation of results.

Usually, reputation is used as the main (or unique) measure for trust, although more aspects can be taken into account and so it is in some of the solutions proposed. Some of the analyzed models are ServiceTrust [2], RateWeb [3] and a model defined in the COIN project [4]. There are also other former models but which propose interesting ideas such as NICE [5], REGRET [6] or Afras [7].

While Afras is a system based on fuzzy logic representing the degree of satisfaction (which aggregates opinions depending on weights, giving more importance to the latest ones), NICE is more oriented to peer-to-peer (P2P) networks, where each node keeps a cookie (with value between 0 and 1) with information about performed transactions. When a node has no direct information about another one, it infers it using others' cookies by means of an oriented weighted graph. In the case of REGRET, a weighted average is performed based on the time and it uses concepts semantically related for calculating the trust.

Latest models are more complex in which refers to the calculation. While ServiceTrust aggregates users' opinions using the function obtained from probabilistic distribution (based on a normal distribution), RateWeb is focused on a weighted average taking into account users' credibility and using a Hidden Markov Model as a way for inferring expected opinions when the number of opinions received is low, since the result would not be very accurate.

These models lack the usage of more information sources (not only opinions, but more aspects) and their robustness is limited to the ability to filter some of the inputs received, without performing a deep analysis of the data being interchanged by all the parties involved. The main challenge is to improve the accuracy of the trust calculation while the robustness of the model is increased at the data level, in order to resist any malicious attack trying to alter the trust calculation.

As a result, [4] proposes a complex and wide model for services, using several heterogeneous aspects which have to be calculated in different ways (especial averages, exponential smoothing, fuzzy logic, etc…). It uses a fuzzy set as the mean to represent the aspects values (see Figure 1) and the main aspects are aggregated by using a weighted average which is based on a weight given by the model administrator (as a way to customize it), time and semantic relationships between aspects (if an aspect is directly related with another one updated recently, it is expected to be more important and represent the actual behavior of the service).

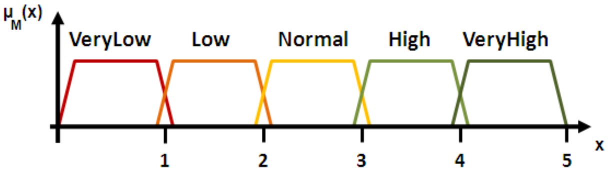


Figure 1. Linguistic terms and their membership function.

While [4] presented a global vision of the trust model and global aggregation functions, this paper presents some of the concrete calculations (closer to the implementation level) performed for some of the aspects, which are based on statistical hypothesis tests, as a tool to exploit the data available from monitoring tools, from external platforms and from users.

III. Main Objectives

As it is possible to obtain a lot of information from different sources, there is the opportunity to exploit it in an adequate way in order to determine the concrete value for some aspects which affect the global trust of a service. These aspects will be key enablers for the improvement of accuracy in trust calculation.

There are two main objectives which are fulfilled by using statistical tests in the trust model:

- Calculate aspects which are not available in other models, as a way to have a better idea of the behavior of the service and its provider, and as a way to evaluate aspects which are important for users but which are very difficult to control and monitor.
- Improve the robustness of the model at the data level, as tests give a good idea of what is going on at a general level with the service, taking many data from several sources, improving the accuracy of the model at the same time.

In order to achieve these principles, next sections propose the way to exploit the information available, which will be provided from a monitoring tool (available in the implementation), from users (gathered through simple forms) and from external platforms, which may be organized in federations as well.

IV. Agreements Fulfillment

The model presented in [4] proposes an aspect which compares monitored values with those agreements made, in order to determine whether a service is behaving as expected, according to contractual commitments.

The idea is to evaluate whether those agreements related to Quality of Service (QoS) parameters (such as response time, availability, etc.) and Trust Level Agreements (TLAs) are fulfilled as expected when users interact with the service. The principle of the aspect is that a service provider cannot be trusted if the agreements are not being fulfilled and the services provided are not stable enough, as expected in business environments.

There are two main parameters to take into account, as defined in the previously mentioned model:

- The stability of the service – Sometimes, the service may behave very well but others it may be behaving wrong (because of technical problems or a wrong development), so it is necessary to guarantee that the service is stable in general, with respect to the agreed QoS and published non-functional properties.
- The fulfillment of agreements – It is mandatory to compare the measured values, obtained from monitoring tools, with the agreed ones. Not only the last value measured will be used, but some measures as well, as a way to give a general view of the service behavior.

There are concrete statistical hypothesis tests which are designed to analyze the variance and average of a set of measured values. Moreover, the inputs of these tests are continuous variables, instead of categories, a fact which facilitates their usage in the presented context.

While the first test is applied for confirming the stability in the variance (a requirement for applying the second test), the second one compares the average of measured values with the expected value for each parameter.

A. Stability Analysis

One of the questions to be answered is whether the service uses to behave in a similar way each time it is invoked. As a service in normal operation is expected to behave always in similar conditions (if no external factors alter its status), the measures obtained from monitoring tools are expected to represent a 'population' following a normal distribution. Consequently, it is possible to perform a Chi-square test in order to determine whether the variance of the measures taken has a pre-determined value, by taking into account each parameter monitored during the service usage.

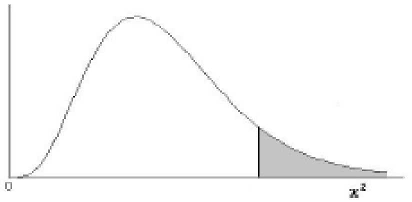


Figure 2. Representation of Chi-square test (Chi-square distribution).

It is possible to determine if the variance is too high by using the null hypothesis H_0: $\delta^2 \leq \delta_0^2$, being δ_0^2 the 5% of the agreed value for the parameter, which means that a variance of 5% in the measured values is acceptable. On the contrary, the alternate hypothesis would represent a variance higher than the 5%. This is a case of unilateral contrast with one degree of freedom.

If the null hypothesis is accepted, then the variance is good, otherwise it is considered bad. One value will be obtained for each parameter in the Service Level Agreements (SLAs) and TLAs with the following equation for one-sample Chi-square test (see [8]).

$$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2} \tag{1}$$

The inputs needed are the expected values for each aspect (for calculating δ value) and the last measures taken by the monitoring tools from each aspect to be evaluated (for obtaining s value). The variable n represents the number of measures taken and used in the test.

B. Agreements Fulfillment Analysis

A key factor for determining the trust in a service is the evaluation of agreements fulfillment by the service and the service provider. As there are contractual commitments about the QoS to be offered (by means of agreed SLAs), it is possible to use information from monitoring tools to determine whether the contracts are being fulfilled.

A way for checking the fulfillment of these agreements is to compare the average of the measured parameters with the agreed value. This is done with a contrast 2-tailed Z-test (see [8]), using the agreed value as the expected mean μ_0.

$$z = \frac{\overline{x} - \mu_0}{(\sigma/\sqrt{n})} \tag{2}$$

One-sample Z-test is a statistical test applicable to populations with well known nuisance parameters, such as variance (which can be easily calculated in this case), and which are expected to vary in a normal distribution.

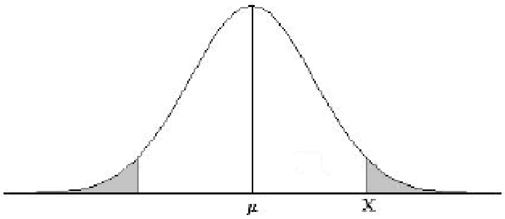


Figure 3. Representation of Z-test (normal distribution).

In this case, the null hypothesis defined is H_0: $\mu = \mu_0$, being μ_0 the agreed value for the parameter, δ will be the variance calculated, and n will depend on the amount of gathered data. The significance level will be 0.05 and the standard deviation will be calculated directly from the measured values during monitoring. This means that the monitored values are expected to have an average of μ_0 with a small variance and very small error which would be accepted. The alternative hypothesis is an average different from μ_0 (H_1: $\mu \neq \mu_0$).

As the Z-test will be calculated once for each parameter under evaluation, there will be a count of not fulfilled parameters, a count of fulfilled parameters and a count of improved parameters (for those values which are better than expected and agreed).

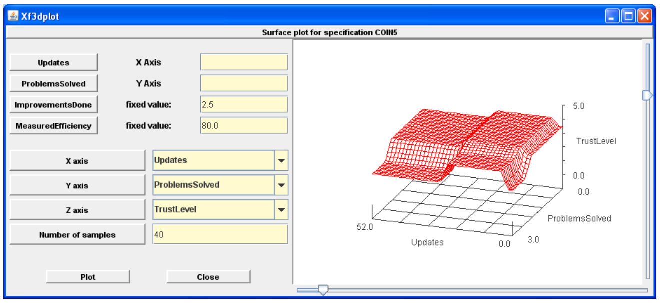
Finally, these counts are taken into account in order to determine how good the service is. In case there are parameters which were not fulfilled, the trust will always be ‘low’ or ‘very low’ (if the variance is too high). In case the agreed values for QoS are fulfilled (or better), the result will be ‘high’ or ‘very high’ (if the service is stable).

V. Release Improvement

According to the model in [4], the aspect ‘Release’ is used for determining how good the maintenance of the service is and how effective new releases are. Some key parameters are taken into account:

- Releases periodicity – The usual time between two releases. A periodicity in releases is good, as it means that the service is maintained, although too many releases could be considered bad.
- Releases successfulness – Measure of how better the service is after the last release, according to the parameters measured. It is represented by the number of improvements in non functional aspects which can be measured.
- Problem solved – Number of problems solved in new releases in comparison to previous releases.
- Functionalities added/improved – Number of functionalities improved or added to the service with the new release.

Although these parameters are aggregated by using a fuzzy model which provides the trust value (one of the simulations is showed in Figure 4), it is very useful to apply a statistical test for performing the comparison between previous and current release, according to measured values through monitoring (represented by the parameter "Release successfulness"). This requires letting the service be used for some time before the test can be performed, as it is necessary to gather some data by monitoring the service behavior during some invocations.



X Axis:

Y Axis:

fixed value: 2.5

fixed value: 80.0

X axis: Updates

Y axis: ProblemsSolved

Z axis: TrustLevel

Number of samples: 40

Figure 4. Fuzzy simulation for the 'Release' aspect.

A. *Release Successfulness Analysis*

This parameter requires confirming that the service has been improved by checking improvements in parameters such as response time, availability and robustness. This can be measured by comparing the averages of measured data for each parameter under evaluation. If measures have improved for, at least, most of the parameters, it can be considered that the release really was successful in the sense that the improvements in the service are clearly observable.

The way to compare averages is performing a Student's t test (to be more concrete, an unpaired t-test, as there are no correspondences between requests) between old monitored measures and new ones (taken after the deployment of the new release).

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1X_2} \cdot \sqrt{\frac{2}{n}}} \tag{3}$$

For the implementation, it is necessary to calculate average values for each group (the group with old values and the one with new values, for X_1 and X_2 averages) and the standard deviation for S. As old values can be filtered using only those closer to the release, the size of the samples will be the same, represented by n. The significance level, which will delimit the critical region, will be again 0.05 (as usually recommended).

As the measures come from the same monitoring tools and their nature and source is the same, it is expected that the variance in both cases will be similar, fulfilling the requirements for performing the test.

In this case, the null hypothesis for each test is H_0: "There are no significant differences between the average values of the compared groups" and the alternate hypothesis would be the contrary (H_1: "There are significant differences between the average values of the compared groups"). The first case means that there are not concrete improvements, while the second case requires checking whether the measures taken are better or worse in order to determine the degree of improvement thanks to the new release.

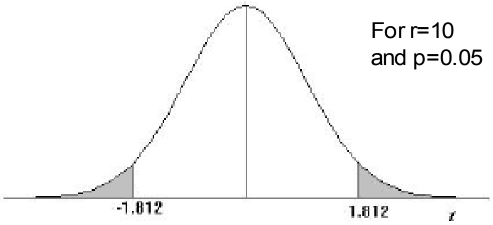


Figure 5. Representation of Student's t test (T Student distribution).

After the test is solved for each measurable non-functional property, the percentage of improved parameters is obtained. For instance, 80% would mean that most of the parameters have been observed to be better than in the previous release. This will be the input for the fuzzy model showed in Figure 4.

B. *Non-Functional Properties Checking*

The model presented in [4] describes an area about service claims regarding functionalities and non-functional properties which are relevant when the service is new and when there is a new release of the service. Although not included currently in the model, there is the possibility to include an analysis about how much the service fulfills the non-functional properties claimed by the service provider.

Mc Nemar's X^2 (see [8]) can be used for comparing claimed Non-Functional Properties (NFPs) and measured NFPs before and after the release. Columns will be 'fulfilled NFPs' and 'not fulfilled NFPs' after the release, while rows will be the same before the release. As usual, the significance level recommended is 0.05.

This means that the null hypothesis would be H_0: "The number of correct claims about NFPs is invariable before and after the release", while the alternate hypothesis would be represented by H_1: "The number of correct claims about NFPs has varied after the release".

In order to calculate it, information contained in the service description can be used as input, as well as the measures taken by the monitoring tools.

As the number of NFPs is not very high and it is not expected to vary, this calculation may not be very significant in the model proposed in [4], but it could be useful in more complex models where there are claims about many parameters.

VI. Opinions and Feedback

One of the essential inputs in any trust model is the usage of information provided by third parties about the element under evaluation. In the case of the model defined in [4], there are two kind of external inputs used: users' feedback and platforms feedback.

In the case of users' feedback, while users execute a business process, they are requested to evaluate a small set of concrete characteristics of the services used in the process. They just need to fill in a very simple form (an eBay like form) and submit it to the Trust Manager.

On the contrary, in the case of platforms feedback, a web service interface is used for requesting information about a service, with the support of an ontology, which is used for mapping aspects of two different platforms. The external platform may send one or more aspects calculated for the service, depending on the information in its trust model.

The approach presented in [4] is based on a weighted average in which received values are more or less important depending on the credibility of the platform or user who provided the value. This credibility is based on two parameters: coincidence in measures and affiliation with the current platform. Both of them will be multiplied for obtaining the final value of credibility.

Cohen's Kappa (see [9]) will be calculated for checking coincidences in measures, comparing measures of one platform and measures of other platform. Depending on the result, it is possible to determine the credibility of an external platform. This is because, as a platform has direct measures about the behavior of a service, if external entities are giving very different feedback it means that different models are being used or malicious information is being received, although it is possible that our monitoring system has some kind of problem. So the weight will be decreased, but the evaluation received will not be ignored at all.

For each parameter measured by both parties, one table is created. Columns will represent results observed by one platform (how many calls were categorized as 'VeryHigh', 'High', 'Medium', 'Low', 'VeryLow') and rows will represent results observed by the platform requesting for information. Then, kappa index will be calculated.

$$k = \frac{p_a - p_r}{1 - p_r} \qquad (4)$$

By applying the formulas for p values defined in [9] to the values in the matrix, the kappa value is obtained. Values provided by each platform are directly requested using the ontology of the trust model in [4].

According to Landis and Koch [10], the level of agreement in measures can be categorized as follows:

TABLE I. AGREEMENT LEVELS

Kappa	Agreement
< 0	Disagreement
0 – 0,2	Slight agreement
0,2 – 0,4	Fair agreement
0,4 – 0,6	Moderate agreement
0,6 – 0,8	Good agreement
0,8 - 1	Very good agreement

The value received from the kappa calculation is directly used for modifying the weight of the data received, by multiplying it to the affiliation (a number between 0 and 1 which represents the historical relationship between the platform and the rater which provided an evaluation). This means that all the weights will be between 0 and 1.

VII. Conclusions and Future Work

Although there are several models defined for calculating the trust associated to a service, most of them are based only in opinions received from users. Moreover, this feedback is used normally by applying weighted measures and, sometimes, probability distributions, but in no cases statistical analysis is exploited for obtaining the trust value.

Used in the proper way, the statistical tests are an interesting tool in order to determine which values are more important or have more sense, supporting as well the robustness against malicious attacks.

But the more interesting usage of statistical tests comes from their utility for calculating other aspects which enrich the trust model with more information, such as those aspects mentioned in the paper (agreements fulfillment and release analysis). These aspects are not included in other models and statistical tests provide the means to study and predict the service behavior thanks to the information gathered during services invocation.

As the model were the mentioned aspects are included is based on a common fuzzy set (representing data in categories) and as values obtaining for monitoring are data of continuous nature, the number of test to be applied is not restricted, by performing adequate normalization.

Next actions are to analyze how to use other statistical tests in the model as a way to improve it. For instance, in the case of feedback analysis, Fleiss' Kappa could be used for providing a more accurate idea about the trustworthiness and weight of service raters (users or external platforms), detecting whether there is some consensus in the behavior of a service.

Finally, as the model defined in [4] uses fuzzy rules in one of its rounds, it is planned to perform some analysis which will provide information about correlations and relationships between different aspects in the model. This way, rules for robustness will be built according to the results obtained, as a way to provide a more robust model.

Acknowledgment

This research was supported by the COIN project (http://www.coin-ip.eu/) and has been partly funded by the European Commission's IST priority of the 7th Framework Programme under contract number 216256.

References

[1] Jøsang, A., Ismail, R., and Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. 43, 2, pp. 618-644. Mar. 2007

[2] He, Q., Yan, J., Jin, H., and Yang, Y. “ServiceTrust: Supporting Reputation-Oriented Service Selection”. In Proceedings of the 7th international Joint Conference on Service-Oriented Computing (Stockholm, November 24 - 27, 2009). L. Baresi, C. Chi, and J. Suzuki, Eds. Lecture Notes In Computer Science, vol. 5900. Springer-Verlag, Berlin, Heidelberg, 269-284.

[3] Malik, Z., Akbar, I., and Bouguettaya, A. “ Web Services Reputation Assessment Using a Hidden Markov Model”. In Proceedings of the 7th international Joint Conference on Service-Oriented Computing (Stockholm, November 24 - 27, 2009). L. Baresi, C. Chi, and J. Suzuki, Eds. Lecture Notes In Computer Science, vol. 5900. Springer-Verlag, Berlin, Heidelberg, 576-591.

[4] Nieto, F. J. “A Trust Model for Services in Federated Platforms”, Vol. 76 of Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, 2011, pp. 118–131.

[5] S. Lee, R. Sherwood, and B. Bhattacharjee. “Cooperative Peer Groups in NICE”. In IEEE Infocom, San Francisco, CA, Apr. 2003.

[6] Sabater, J. and Sierra, C. “REGRET: A reputation model for gregarious societies”. In Proceedings of the 4th Int. Workshop on Deception, Fraud and Trust in Agent Societies, in the 5th Int. Conference on Autonomous Agents (AGENTS’01), pages 61-69, Montreal, Canada, 2001.

[7] Carbo, J., Molina, J., and Davila, J. “Comparing predictions of SPORAS vs. a Fuzzy Reputation Agent System”. In: 3rd International Conference on Fuzzy Sets and Fuzzy Systems, Interlaken, 2002. pp. 147—153.

[8] “NIST/SEMATECH e-Handbook of Statistical Methods”, http://www.itl.nist.gov/div898/handbook/, 2010 (accessed June 2011).

[9] Sim, J.; Wright, C. C.; "The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements". Physical Therapy 85 (3): 257–268 (2005).

[10] Landis, J. R. and Koch, G. G. "The measurement of observer agreement for categorical data" in Biometrics. Vol. 33, pp. 159–174 (1977).

ESARC - Enterprise Services Architecture Reference Cube for Capability Assessments of Service-oriented Systems

Alfred Zimmermann
Reutlingen University
Software Architecture Research Group
Reutlingen, Germany
alfred.zimmermann@reutlingen-university.de

Gertrud Zimmermann
ZIMMERMANN UND PARTNER
Enterprise Software Architecture Research
Pfullingen, Germany
gertrud.zimmermann@online.de

***Abstract* – An original ESARC-Enterprise Services Architecture Reference Cube for supporting evaluation and optimization of service-oriented architectures is introduced. Current approaches for assessing architecture quality and maturity of service-oriented enterprise software architectures are rarely validated and were intuitively developed, having sparse reference model, metamodel or pattern foundation. Cyclic assessments of complex service-oriented systems and architectures should produce convergent and comparable evaluation results. Today architecture evaluation findings are hardly comparable. This is a real problem in cyclic evaluations of advanced architecture quality concepts to get a stable foundation for introducing service-oriented enterprise architectures for adaptive systems. Our idea and contribution is to extend existing enterprise and software architecture reference models and maturity frameworks to accord with an integral enterprise architecture reference model approach. We have applied our service-oriented ESARC in several assessment workshops with global vendors of service-oriented platforms. This experience provides the base for further investigations and improvements of our approach. ESARC provides for both cyclic architecture quality assessments and for the architecture construction and optimization a standardized and normative classification scheme of important architecture artifacts for service-oriented enterprise systems.**

Keywords – Service-oriented Architecture; Enterprise Architecture; ESARC; Reference Architecture; Architecture Patterns; Architecture Capability and Maturity Assessments.

I. INTRODUCTION

Since recent years innovation oriented companies have introduced service-oriented computing paradigms and combine this with traditional information systems. Typical service-oriented technologies include systems following a service-oriented architecture (SOA). Service-oriented systems close the business - IT gap by delivering efficiently appropriate business functionality and integrating legacy systems with standard application platforms. Our approach is to investigate the practical use of the SOA ability of standard platforms in commercial use [1] for members of the SOA Innovation Lab, which is a major innovation and research network in Germany and Europe.

In assessing the quality of implemented SOA vendor platforms and the integral architecture of service-oriented enterprise systems, we were faced with the problem of not real comparable evaluation findings from consecutive (cyclic) assessments of heterogeneous systems. Our previous assessment findings were done without an architecture reference model. This causes that multiple evaluations of enterprise systems with service-oriented architectures were blurry and hardly comparable within a series of consecutive architectural tests and therefore produced less meaningful assessment results. The aim of our research is to enhance analytical instruments for cyclic evaluations of business and system capabilities of different service-oriented platforms and enterprise systems.

The hypothesis in our current research paper for the ESARC is:

1. ESARC - Enterprise Services Architecture Reference Cube is an effective concretization of the TOGAF [2] framework and other seminal work on enterprise architectures [3], service-oriented reference models and software architectures [4], [5], [6], and defines useful architecture artifacts with their main relationships.
2. ESARC provides a useful foundation of a reference structure for metamodel-based capability assessments of service-oriented systems and their architecture [1], [7], as well as for architecture assessment patterns [8] from previous work.

We are reporting about a novel holistic approach of the ESARC – the Enterprise Services Architecture Reference Cube, which helps enterprise and software architects to define and structure their evaluation object - the service-oriented enterprise and software architecture - in a standard way. In order to specify our innovative enterprise and software architecture assessment method, we used a metamodel-based approach for capability evaluations of architecture elements and their main relationships. For this purpose we have extended, integrated and adapted elements from convergent architecture methods, patterns, related standards and reference models from the state of art.

In the following Section II, we introduce base and seminal related work on reference models, reference architectures and architecture patterns, as well as open architecture standards, frameworks, and service-oriented architecture maturity models. In Section III, we present the main view of our original developed ESARC architecture reference model. Section IV mentions results from our method validation of ESARC from assessments of four major SOA vendor platforms. Finally, Section V summarizes our conclusions and mentions some ideas from current research and for future work.

II. RELATED WORK

Our research is based on following formal architecture concepts from [9] and their relationships: software architecture, reference architecture, reference model, and architecture patterns. A reference model for SOA [4] is a generic fundamental model as in [9] that embodies the basic idea and provides a decomposition of functionality of a given problem, together with the data flow between elements. The reference model contains an abstract technology agnostic representation of the elements and their relationships, showing the interactions between basic concepts. The concept of reference architecture [9] and [5], [6] is the result of a mapping of an architecture reference model to software elements and contains the related data flow between them.

Architecture patterns are representations of a set of architectural constraints for architecture elements and their relation types. Architecture patterns [9], and [10], [11] show quality attributes and represent known solutions for a given problem. An architecture pattern records the architecture decisions taken by many architects in order to resolve a particular architecture problem. Patterns are human readable structures of text and graphics showing a standardized and repeatable way to derive a solution from a specified problem in a specific context. Our developed and practically validated pattern catalog [8] for quality patterns of enterprise software architectures relies originally on our previous developed service-oriented architecture maturity framework [7] for assessing architecture capabilities and maturity of service-oriented enterprise systems.

The Architecture Tradeoff Analysis Method (ATAM) [15] is a foundation method for our specific architecture evaluations of service-oriented enterprise systems. A seminal work used in the preparation of our service-oriented architecture assessments is [16], which provides concrete guidelines for the design of our questionnaire as in [1].

Service-oriented architecture SOA [12] is the computing paradigm that utilizes services as fundamental flexible and interoperable building blocks for both structuring the business and for developing applications. SOA promotes a business oriented architecture style, based on best of breed technology of context agnostic business services that are delivered by applications in a business focused granularity. To provide agile composition of services within a worldwide environment and to enable flexible integration of published and discovered components, SOA uses a set of XML-based standards like WSDL, SOAP, UDDI, and others. A main innovation introduced by SOA is that business processes are not only modeled. Business process models are used in a more mature way consistently within a Model Driven Architecture (MDA) approach to generate new and agile orchestrations or compositions of web services based on process diagrams. Early definitions of SOA were technology focused and the differences between SOA and web services were often blurred. SOA technologies emerged due to the expansion of the Internet technology during the last years and produced abundance specifications and standards as in [4], [5], [6], and [13], which are developed by open standard organizations like W3C, OMG, OASIS, and The Open Group. The perspective of a service development process is offered by [14] and [11].

Our architecture reference model ESARC relates closely to SOAMMI, which is our previous designed maturity framework for evaluation of enterprise and service-oriented product architectures. Unfortunately most of existing SOA and EA maturity models lack a clear metamodel base. Therefore we have extended CMMI [17] in our previous research, which is a framework for assessments of software processes, and transformed it into a specific framework for the assessment of the maturity of service-oriented enterprise and software architectures [1] and [7]. Therefore we have combined and extended CMMI with architecture quality criteria from current architecture frameworks and architecture maturity models. In particular we use TOGAF [2] as a basic structure for enterprise architecture, spanning all relevant levels. In addition, we have cross checked and – if appropriate - extended our metamodel with supporting elements from known maturity models.

The Architecture Capability Maturity Model (ACMM) [18] framework, which is included in TOGAF, was originally developed by the US Department of Commerce. The main scope of ACMM is the evaluation of enterprise architectures in internal enterprise architecture assessments. The goal of ACMM assessments is to enhance enterprise architectures by identifying quantitative weak areas and to show an improvement path for the identified gaps of the assessed architecture. The ACMM spans six maturity levels and defines nine specific architecture elements.

The SOA Maturity Model in [19] considers the following multidimensional aspects of a SOA: scope of SOA adoption, SOA maturity levels - to express architecture capabilities, SOA expansion stages, SOA return on investment, as well as SOA cost effectiveness and feasibility. The scope of SOA adoption in an enterprise is differentiated by following levels: intra-department or ad hoc adoption, inter-departmental adoption on business unit level, cross business unit adoption, and the enterprise level, including the SOA adoption within the entire supply chain.

The SOA Maturity Model from Sonic [20] distinguishes five maturity levels of a SOA, and associates them - in analogy to a simplified metamodel of CMMI - with key goals and key practice. Key goals and key practices are reference points in SOA maturity assessments.

The SOA Maturity Model of ORACLE in [21] characterizes in a loose correlation with CMMI five different maturity levels and associates them with strategic goals and tactical plans for implementing SOA. Additional capabilities of a SOA are referenced with each maturity level: Infrastructure, Architecture, Information & Analytics, Operations, Project Execution, Finance & Portfolios, People & Organization, and Governance.

III. ESARC – Enterprise Services Architecture Reference Cube

ESARC is an original abstract architecture reference model which defines an integral view for main interweaved architecture types. ESARC was derived primarily from state of art architecture frameworks like TOGAF [2], essential [3], the service model of ITIL, and from resources for service-oriented computing [11], [12], [5]. The aim of the ESARC architecture reference model is to be universally applicable in different cyclic repeatable architecture evaluations and structural optimizations of enterprise and software architectures. ESARC abstracts from a concrete business scenario or technologies.

The Open Group Architecture Framework (TOGAF) [2] is the current standard for enterprise architecture and provides the basic blueprint and structure for the service-oriented enterprise software architecture domains of ESARC: Architecture Governance, Architecture Management, Business & Information Architecture, Information Systems Architecture, Technology Architecture, Operation Architecture, and Service Architecture.

The formal foundation for ESARC, as detailed in [4], [5], and [6], is an abstract representation of standardized architecture building blocks in a layered acyclic relationship. The layer semantics implies that the basic layers are prerequisites for higher architecture layers. At a higher granularity, all architecture domains are parts of the holistic architecture composition framework of ESARC.

The ESARC – Enterprise Services Architecture Reference Cube unifies orthogonal architecture domains into aligned architecture views, which yield an aid for examination, comparison, classification and quality rating of different architecture aspects. ESARC is our holistic definition of a full service-oriented architecture used both for assessing and optimization of service-oriented product lines and for families of application systems. Our unifying perspective of service-oriented enterprise systems integrates and helps to align business and the technology aspects.

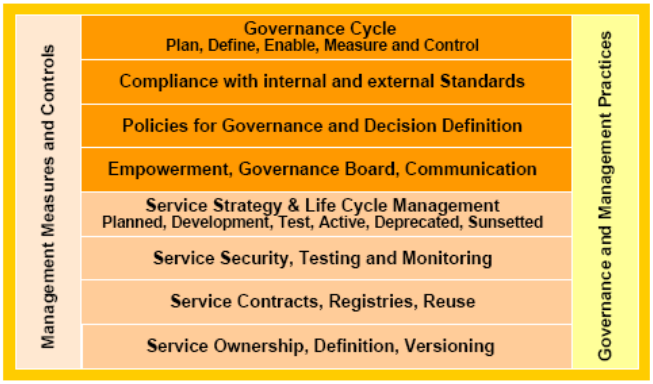

Management Measures and Controls
Governance Cycle
Plan, Define, Enable, Measure and Control
Compliance with internal and external Standards
Policies for Governance and Decision Definition
Empowerment, Governance Board, Communication
Service Strategy & Life Cycle Management
Planned, Development, Test, Active, Deprecated, Sunsetted
Service Security, Testing and Monitoring
Service Contracts, Registries, Reuse
Service Ownership, Definition, Versioning
Governance and Management Practices

Figure 1. ESARC – Architecture Governance and Management.

The main types of enterprise software architectures like Business & Information Architecture, the Information Systems Architecture, and the Technology Architecture are organized by the Architecture Governance and Management framework. Architecture Governance as in Figure 1 conforms to the SOA Governance Framework in [13] and defines and maintains the Architecture Governance cycle.

The Architecture Governance cycle sets the abstract governance frame for concrete architecture activities within the enterprise software or a product line development. The Architecture Governance cycle specifies the following management activities: plan, define, enable, measure, and control. The second aim of Architecture Governance is to set rules for architecture compliance with internal and external standards. Policies for governance and decision definition are set, to allow a standardized and efficient process for architecture decisions within the enterprise architecture organization. Because enterprise and software architects are acting on a sophisticated connection path coming from business and IT strategy to the architecture landscape realization of interrelated business domains, applications and technologies, Architecture Governance has to set rules for empowerment of software people, define the structures and procedures of an Architecture Governance Board, and set rules for communication.

Benefits from well organized architecture governance (adapted from [2]) are: transparency of accountability, informed delegation of authority, controlled risk management, protection of the existing asset base through maximizing reuse of existing architectural components, proactive control, monitoring, and management mechanisms, value creation through monitoring, measuring, evaluation, and feedback, increased visibility of decision-making in supporting internal processes and external requirements, and greater shareholder value. The enterprise architecture increasingly represents the core intellectual property of the enterprise systems. It is a precondition for an effective business and system integration with existing processes and methodologies and adds control capabilities.

With specifications from Architecture Governance we define our main Architecture Management procedures for service-oriented enterprise software architectures: service strategy and life cycle management of software and system architecture artifact's state, service security, service testing and monitoring, service contracts, registries, service reuse, service ownership, definition, and versioning.

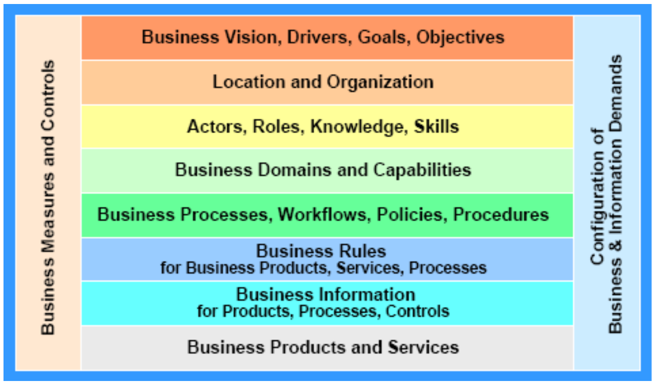

Business Measures and Controls
Business Vision, Drivers, Goals, Objectives
Location and Organization
Actors, Roles, Knowledge, Skills
Business Domains and Capabilities
Business Processes, Workflows, Policies, Procedures
Business Rules
for Business Products, Services, Processes
Business Information
for Products, Processes, Controls
Business Products and Services
Configuration of
Business & Information Demands

Figure 2. ESARC – Business & Information Reference Architecture.

The ESARC - Business & Information Reference Architecture in Figure 2 defines the link between the enterprise business strategy and the resulting business and information design for supporting strategic initiatives. The Business & Information Reference Architecture provides a single source and comprehensive repository of knowledge from which corporate initiatives will evolve and link. This knowledge is model-based and is an integrated enterprise model of the business, which includes the organization and the business processes. The Business & Information Reference Architecture opens a connection to IT infrastructures, systems, as well as to software and security architectures. It provides integration capabilities for IT management, software engineering, service & operations management, and process improvement initiatives. The Business & Information Reference Architecture defines and models the business and information strategy, the organization, and main business requirements for information systems: key business processes, business rules, business products, and business control information.

The ESARC – Information Systems Reference Architecture in Figure 3 provides an abstract blueprint for the individual application architecture to be deployed. It adds specific interactions and specifies relationships to the core business processes of the organization. The OASIS Reference Model for Service Oriented Architecture [4] is an abstract framework which guides our ESARC reference architecture. The ESARC defines the abstract model for specific applications architectures and implementations.

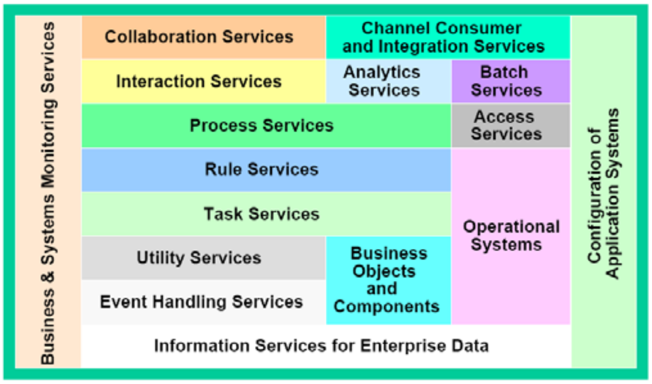


Figure 3. ESARC - Information Systems Reference Architecture.

In our ESARC – Information Systems Reference Architecture we have differentiated layered service types. The information services for enterprise data can be thought of as data centric components [14], providing access to the persistent entities of the business process. The capabilities of information services combine both elementary access to CRUD (create, read, update, delete) operations and complex functionality for finding/searching of data or complex data structures, like data composites or other complex-typed information. Close to the access of enterprise data are context management capabilities, provided by the technology architecture: error compensation or exception handling, seeking for alternative information, transaction processing of both atomic and long running and prevalent distributed transactions. Information services [14] and their related data architecture [2] are core company assets and should be close and centrally managed for reuse. Task services implement business capabilities related to specific actions of the business process. Task services could be own or third-party services. Usually task services don't manage state information directly, but work in cooperation with information services. The access to information services follows an acyclic graph - from top to down layers.

From [11] and [5], [6], [12] result important design rules for task services. Operations of task and entity services shouldn't have any knowledge about their process or interactive usage context. Task service operations [14] should be independent from users and sessions and should only implement business functionality. Authorization checks should be done outside of the business operations. Task and information services should use a transactional context, but their operations shouldn't implement by their own transactions. Task service operations should be usable both in batch and in online system transactions. Task services are used in process services - as multiple composites of services and should therefore be centrally managed high reusable assets. Rule services provide knowledge representation and processing capabilities for adaptable business product and business services. Rule services provide in addition flexible controls for agile business processes.

Process services [14] are long running services which compose task services and information services into workflows, to implement the procedural logic of business processes. Process services can activate rule services, to swap out a part of the potentially unstable gateway-related causal decision logic. Process services are frontend by interaction services or by specific diagnostic service and process monitoring services. Often process services manage distributed data and application state indirectly, by activating task and information services. Process services participate in atomic transactions only when they are activated from batch services. When processes services participate in human interaction workflows, they have to support long-running transactions where compensation of possible errors or exceptions happens in the business logic.

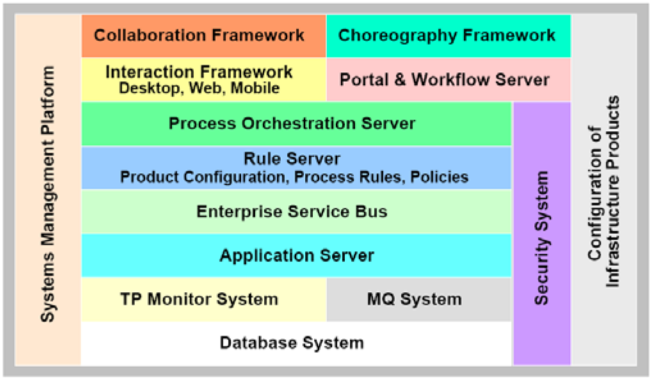

Systems Management Platform
Collaboration Framework
Choreography Framework
Interaction Framework Desktop, Web, Mobile
Portal & Workflow Server
Process Orchestration Server
Rule Server Product Configuration, Process Rules, Policies
Enterprise Service Bus
Application Server
TP Monitor System
MQ System
Database System
Security System
Configuration of Infrastructure Products

Figure 4. ESARC - Technology Reference Architecture.

The ESARC – Technology Reference Architecture in Figure 4 describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing, and standards. The layers of the ESARC – Technology Reference Architecture and the layers of the ESARC – Information Systems Reference Architecture correspond to each other. The database system is the vendor supported database management system for handling enterprise data. We have included in our architecture stack the TP-Monitor system and have associated it with the database system. Optionally we have integrated a messaging system. The application server is the container environment architecture for objects and enterprise components. The enterprise service bus uses a flexible and standard-based messaging mechanism to interconnect services on a process and service execution platform. On top of the service bus we have placed the rule server, which is able to represent business rules and operate rule processing by building dynamic inference chains. The process orchestration server executes process services by calling suitable services of earlier mentioned types. For running interaction and collaboration services additional infrastructures are included in our stack of the technology reference architecture model: interaction frameworks, portal servers, workflow engines, and nowadays collaboration frameworks. Security services are part of an integral framework-based security system of standards and components and are impacted by mentioned services and distributed service technologies.

IV. Validation and Results

Architecture assessments need to address the key challenges for companies during the built-up and management of service-oriented architectures in heterogeneous IT environments. Assessments of the SOA ability of standard software packages can be viewed additionally as a mean to engage with vendors on relevant challenges of SOA in practical use.

The basic structure of a working example is our questionnaire [1], which was based on our currently reported ESARC for the assessment and optimization of architecture artifacts. Our questionnaire is also close associated with our previous developed SOAMMI – a service-oriented enterprise architecture maturity framework [7], [1] and on adapted elements from [16]. Detailed working examples of the ESARC Reference Model, the SOAMMI Maturity Model and our Architecture Capability Patterns in action can be found in our current research paper [22].

We have synthesized the following key findings that highlight our view on the actual SOA ability of a standard platform across vendors:

SOA experiences: Even though SOA has been a topic for vendors for years now, there are no major SOA implementations that include standard software systems. Most cases have the quality of a proof of concept, often focusing on GUI integration, instead of deep functional integration. There seems to be a gap between those SOA capabilities that are offered and those which can be actually used in a SOA.

Architecture strategy management: SOA is seen as an important part of overall strategy with no alternative in the long term. All vendors have developed SOA strategies and have integrated it into their product roadmap. In most cases, SOA enablement is a mandatory requirement for the development of new functionality.

Business Services: Vendors offer solution maps that describe the functionality in terms of services and have developed methods to find existing services to a given requirement. In addition, vendors are developing solution scenarios, which offer not just the individual service but a complete set of processes that implement a business solution.

Business product dependencies: Vendors have invested substantially in SOA, but in many cases, SOA has been only applied as wrapping of existing systems, without changing the core of the application. This means that business services are tightly coupled and therefore inflexible. Often dependencies between services were complex and could be ambiguous for the service composition.

SOA deployment units: No vendor offers licenses that allow the usage of individual services instead of the whole system. This means that users still have to purchase the whole application, which hinders a best of breed approach for composite applications.

SOA methods: There is a rich offering for methods for governance, implementation guidelines, etc. for SOA available. SOA is not just seen as the technical implementation, but rather as an engineering discipline that goes beyond service interfaces.

Security, ESB, ESR, service monitoring: Industry standards are implemented within the standard software, but standards like SAML leave room for interpretation. This makes it difficult to integrate solutions across several standard platforms, which is a requirement for most users.

SOA tools: All standard platform providers have added tool suites to their portfolio that support SOA development. The integration of these tools within development layers and across platforms is still not completely solved.

In summary, there are still obstacles to apply standard software in a heterogeneous SOA environment. Often, a vendor's SOA approach is specific to the vendor. For example, each vendor has structured business functionality - a business domain map – defined and described in an individual way. However these business domain maps are vendor specific and often do not correlate with company specific domain maps. Vendors also often use specific semantics and data models and have incompatible technologies (ESB, repository) that do not integrate seamlessly into overall heterogeneous landscapes.

For most vendors, products are only SOA *enabled*. This means that SOA is implemented as wrapper around existing interfaces, and the internal structure is still monolithic. This typically results in a very granular and technical view (e.g., over 3.000 services) that is difficult for the user to identify and comprehend, and therefore to implement. In addition, there are many dependencies between services that often require certain modules to be implemented and populated with data, before services from other domains can be used.

V. Conclusion and Future Work

Our original approach for architecture evaluation and optimization of service-oriented enterprise software architectures is based on ESARC - a special architecture reference model, an associated architecture metamodel and on architecture patterns. In our research we have motivated the necessity to extend both existing architecture reference models and service-oriented maturity models to accord to a clear metamodel approach due to the well understood and verified CMMI model. Our approach provides a sound basis from theory for practical evaluations of service oriented standard platforms in heterogeneous environments with four major global acting technology vendors. Future work has to consider conceptual work on both static and dynamic architecture complexity, and in connecting architecture quality procedures with prognostic processes on architecture maturity with simulations of enterprise and software architectures. Additional improvement idea deals with patterns for visualization of architecture artifacts and architecture control information to be operable on an architecture management cockpit. To improve semantic-based navigation within the complex space of EAM-visualization and service-oriented enterprise software architecture management and we are working on ontology models for the ESARC – The Enterprise Software Architecture Reference Cube.

Acknowledgement

Many thanks to the SOA Innovation Lab initiative from Germany and Europe, a research and innovation network of information technology applicators, the technology vendors IBM, SAP, ORACLE, Microsoft, the associated consulting firms and scientific network for providing us the context for our research.

References

[1] H. Buckow, H.-J. Groß, G. Piller, K. Prott, J. Willkomm, and A. Zimmermann, “*Analyzing the SOA-ability of Standard Software Packages with a dedicated Architecture Maturity Framework*”, EMISA 2010: October 7– 8, 2010 - Karlsruhe, Germany, GI-Edition - Lecture Notes in Informatics (LNI), P-172, 2010, pp. 131-143.

[2] TOGAF “*The Open Group Architecture Framework*” Version-9, The Open Group, 2009.

[3] *Essential Architecture Project*, http://www.enterprise-architecture.org, last access: June, 19th, 2011.

[4] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, OASIS “*Reference Model for Service Oriented Architecture*” 1.0, OASIS Standard, 12 October 2006.

[5] J. A. Estefan, K. Laskey, F. G. McCabe, and D. Thornton, OASIS “*Reference Architecture for Service Oriented Architecture*” Version 1.0, OASIS Public Review Draft 1, 23 April 2008.

[6] J. A. Estefan, K. Laskey, F. G. McCabe, and D. Thornton, OASIS “*Reference Architecture Foundation for Service Oriented Architecture*” Version 1.0, OASIS Committee Draft 02, 14 October 2009.

[7] A. Zimmermann, “*Method for Maturity Diagnostics of Enterprise and Software Architectures*”, in A. Erkollar (Ed.) ENTERPRISE & BUSINESS MANAGEMENT, A Handbook for Educators, Consulters and Practitioners, Volume 2, Tectum 2010, ISBN 978-3-8288-2306-8, pp. 129-172, 2010.

[8] A. Zimmermann, E. Ammann, and F. Laux, “*Pattern Catalog for Capability Diagnostics and Maturity Evaluation of Service-oriented Enterprise Architectures*”, PATTERNS 2010 - The Second International Conferences on Pervasive Patterns and Applications, November 21-26, 2010 - Lisbon, Portugal, IARIA Proceedings of the PATTERNS 2010 Conference, pp. 13-19, 2010.

[9] L. Bass, P. Clements, and R. Kazman, “*Software Architecture in Practice”*, Second Edition, Addison Wesley, 2003.

[10] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, „*Pattern-oriented Software Architecture“*, Wiley. 1996.

[11] T. Erl, “*SOA Design Patterns*”, Prentice Hall. 2009.

[12] T. Erl, “*Service Oriented Architecture”* Prentice Hall, 2005.

[13] The Open Group “*SOA Governance Framework*”, August 2009.

[14] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.P. Richter, M. Voß, and J. Willkomm, „*Quasar Enterprise*“ dpunkt.verlag, 2008.

[15] R. Kazman, M. Klein, and P. Clements, *“The Architecture Tradeoff Analysis Method (ATAM)”*, CMU/SEI-2000-TR-004, Carnegie Mellon University, Software Engineering Institute, 2000.

[16] P. Bianco, R. Kotermanski, and O. Merson, *“Evaluating a Service-Oriented Architecture”*, CMU/SEI-2007-TR-015, Carnegie Mellon University, Software Engineering Institute, 2007.

[17] CMMI-DEV-1.3 2010 “*CMMI for Development, Version 1.3*” Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2010-TR-033, 2010.

[18] ACMM, “*Architecture Capability Maturity Model*”, in TOGAF Version 9, The Open Group Architecture Framework, The Open Group, 2009, pp. 685-688.

[19] S. Inaganti and S. Aravamudan, “*SOA Maturity Model*” BP Trends, April 2007, 2007, pp. 1-23.

[20] Sonic “*A new Service-oriented Architecture (SOA) Maturity Model*” http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf, last access: June, 19th, 2011.

[21] Oracle “*SOA Maturity Model*”, http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet, last access: June, 19th, 2011.

[22] A. Zimmermann, H. Buckow, H.-J. Groß, O.F. Nandico, G. Piller, and K. Prott, “*Capability Diagnostics of Enterprise Service Architectures using a dedicated Software Architecture Reference Model*”, IEEE-SCC2011: Washington DC – July 5-10, 2011, to be published.

Deriving Interface Contracts for Distributed Services

Bernhard Hollunder
Department of Computer Science
Furtwangen University of Applied Sciences
Robert-Gerwig-Platz 1, D-78120 Furtwangen, Germany
Email: hollunder@hs-furtwangen.de

***Abstract*—Software components should be equipped with well-defined interfaces. With *design by contract*, there is a prominent principle for specifying preconditions and postconditions for methods as well as invariants for classes. Although design by contract has been recognized as a powerful vehicle for improving software quality, modern programming languages such as Java and C# did not support it from the beginning. In the meanwhile, several language extensions have been proposed such as Contracts for Java, Java Modeling Language, as well as Code Contracts for .NET. In this paper, we present an approach that brings design by contract to distributed services. To be precise, contracts included in the implementation of a Web service will be automatically extracted and translated into a so-called contract policy, which will be part of the service's WSDL. Our solution also covers the generation of contract-aware proxy objects to enforce the contract policy already on client side. The feasibility of our approach has been demonstrated for .NET/WCF services and for Java based Web Services.**

***Keywords*-Interface Contracts for distributed services; Design by contract; WCF; Web services; WS-Policy.**

I. Introduction

Two decades ago, Bertrand Meyer [1] introduced the *design by contract* principle for the programming language Eiffel. It allows the definition of expressions specifying preconditions and postconditions for methods as well as invariants for classes. These expressions impose constraints on the states of the software system (e.g., class instances, parameter and return values) which must be fulfilled during execution time.

Although the quality of software components can be increased by applying design by contract, widely used programming languages such as Java and C# did not support contracts from the beginning. Recently, several language extensions have been proposed such as *Code Contracts* for .NET [2], *Contracts for Java* [3] as well as *Java Modeling Language* [4] targeting at Java. Common characteristics of these technologies are *i*) specific language constructs for encoding contracts, and *ii*) extended runtime environments for enforcing the specified contracts. Approaches such as Code Contracts also provide support for static code analysis and documentation generation.

In this work, we will show how distributed services such as Web services can profit from the just mentioned language extensions. The solution presented tackles the following problem: Contracts contained in the implementation of a Web service are currently completely ignored when deriving its WSDL interface. As a consequence, constraints such as preconditions are not visible for a Web service consumer.

Key features of our solution for bringing contracts to distributed services are:

- simplicity
- automation
- interoperability
- client side support
- feasibility
- usage of standard technologies.

Simplicity expresses the fact that our solution is transparent for the service developer—no special activities must be performed by her/him. Due to a high degree of *automation*, the constraints (i.e., preconditions, postconditions, and invariants) specified in the Web service implementation are automatically translated into equivalent contract expressions at WSDL interface level.

As these expressions will be represented in a programming language independent format, our approach supports *interoperability* between different Web services frameworks. For example, Code Contracts contained in a WCF service implementation will be translated into a WSDL contract policy, which can be mapped to expressions of the Contracts for Java technology used on service consumer side. This *client side support* is achieved by generating contract-aware proxy objects. The *feasibility* of the approach has been demonstrated by proof of concept implementation including tool support.

In order to represent contract expressions in a Web service's WSDL, we will employ *standard technologies*: *i*) WS-Policy [5] as the most prominent and widely supported policy language for Web services, *ii*) WS-PolicyAttachment [6] for embedding a contract policy into a WSDL description, and *iii*) the Object Constraint Language (OCL) [7] as a standard of the Object Management Group (OMG) for representing constraints in a programming language independent manner.

Before we explain our solution in the following sections, we observe that several multi-purpose as well as domain-specific constraint languages have already been proposed for Web services (see, e.g., [8], [9], [10]). However, these papers have their own specialty and do not address important features of our approach:

- Contract expressions are automatically extracted from the service implementation and mapped to a corresponding contract policy.
- Our approach does not require an additional runtime environment. Instead, it is the responsibility of the underlying contract technology to enforce the specified contracts.
- Usage of well-known specifications and widely supported technologies. Only the notions “contract assertion” and “contract policy” have been coined in this work.

To the best of our knowledge, the strategy presented in this paper has not been elaborated yet elsewhere.

The paper is structured as follows. Next we will recall the problem description followed by the elaboration of the architecture and an implementation strategy on abstract level. So-called contract policies will be defined in Section IV. Then we will apply our strategy to Code Contracts for WCF services (Section V) and Contracts for Java for JAX Web services (Section VI). Limitations of the approach will be discussed in Section VII. The paper will conclude with a summary and directions for future work.

II. Problem Description

We start with considering a simple Web service that returns the square root for a given number. We apply Code Contracts [2] and Contracts for Java [3], respectively, to formulate the precondition that the input parameter value must be non negative.

The following code fragment shows a realization as a WCF service. According to the Code Contracts programming model, the static method `Requires` (resp. `Ensures`) of the `Contract` class is used to specify a precondition (resp. postcondition).

```
using System.ServiceModel;
using System.Diagnostics.Contract;

[ServiceContract]
public interface IService {
  [OperationContract]
  double squareRoot(double d);
}

public class IServiceImpl : IService {
  public double squareRoot(double d) {
    Contract.Requires(d >= 0);
    return Math.Sqrt(d);
  }
}
```

WCF service with Code Contracts.

The next code fragment shows an implementation of the square root service in a Java environment. In this example, we use Contracts for Java. In contrast to Code Contracts, Contract for Java uses annotations to impose constraints on the parameter values: `@requires` indicates a precondition and `@ensures` a postcondition.

```
import javax.jws.WebMethod;
import javax.jws.WebService;
import com.google.java.contract.Requires;

@WebService()
public class Calculator {
  @WebMethod
  @Requires("d >= 0")
  public double squareRoot(double d) {
    return Math.sqrt(d);
  }
}
```

Java based Web service with Contracts for Java.

Though the preconditions are part of the Web service definition, they will not be part of the service’s WSDL interface. This is due to the fact that during the deploying of the service *its preconditions, postconditions, and invariants are completely ignored* and hence are not considered when generating the WSDL. This is not only true for a WCF environment as already pointed out in [11], but also for Java Web services environments such Glassfish/Metro [12] and Axis2 [13].

As contracts defined in the service implementation are not part of the WSDL, they are not visible to the Web service consumer—unless the client side developer consults additional resources such as an up to date documentation of the service. But even if there would exist a valid documentation, the generated client side proxy objects will not be aware of the constraints imposed on the Web service. Thus, if the contracts should already be enforced on client side, the client developer has to manually encode the constraints in the client application or the proxy objects. Obviously, this approach would limit the acceptance of applying contracts to Web services.

Our solution architecture overcomes these limitations by automating the following activities:

- Contracts are extracted from the service implementation and will be transformed into corresponding OCL expressions, which are packaged as WS-Policy assertions (so-called contract assertions).
- A contract policy (i.e., a set of contract assertions) will be included into the service’s WSDL.
- Generation of contract-aware proxy objects—proxy objects that are equipped with contract expressions derived from the contract policy.
- Usage of static analysis and runtime checking on both client and server side as provided by the underlying contract technologies.

An important requirement from a Web service development point of view is not only the automation of these activities, but also a seamless integration into widely used

Integrated Development Environments (IDEs) such as Visual Studio, Eclipse, and NetBeans. For example, when deploying a Web service project no additional user interaction should be required to create and attach contract policies.

III. Architecture

The following figure shows the main components of our solution.

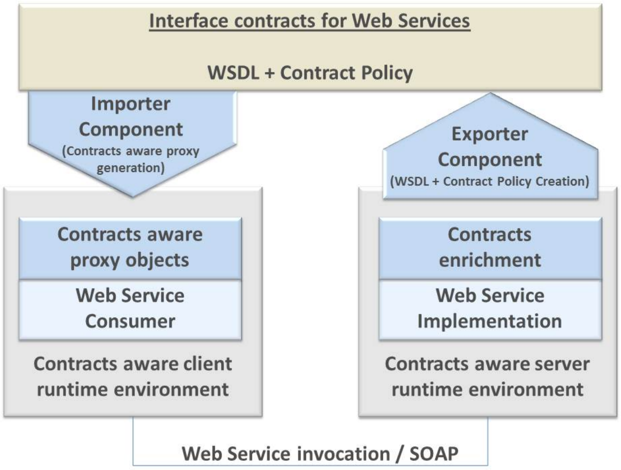

Interface contracts for Web Services
WSDL + Contract Policy
Importer Component
(Contracts aware proxy generation)
Exporter Component
(WSDL + Contract Policy Creation)
Contracts aware proxy objects
Web Service Consumer
Contracts aware client runtime environment
Contracts enrichment
Web Service Implementation
Contracts aware server runtime environment
Web Service invocation / SOAP

Figure 1. Solution architecture.

In short, our approach adopts the *code first strategy* for developing Web services. One starts with implementing the Web service's functionality in some programming language such as C# or Java. We assume that some contract technology is used to enhance the service under development by preconditions, postconditions, and invariants. In Figure 1, this activity is indicated by *contract enrichment*. At this point, one ends up with a contract-aware Web service such as the sample square root service at the beginning of Section II. In order to properly evaluate the contracts during service execution, a contract-aware runtime environment is required. Such an environment is part of the employed contract technology.

The standard deployment of the Web service will be adapted such that a contract policy is created and attached to the WSDL. The *exporter component* performs the following actions:

1) Extraction of contract expressions by inspecting the Web service implementation.
2) Construction of contract assertions and contract policies.
3) Creation of the service's WSDL and attachment of the contract policy.
4) Upload of the WSDL on a Web server.

Note that the contract policy will be part of the service's WSDL and is therefore accessible for the service consumer. Both the WSDL and the contract policy is used by the *importer component* to generate and enhance the proxy objects on service consumer side. The importer component fulfills the following tasks:

1) Generation of the "standard" proxy objects.
2) Translation of the contract assertions contained in the contract policy into equivalent expressions of the contract technology used on service consumer side.
3) Enhancement of the proxy objects with the contract expressions created in the previous step.

Note that service consumer and service provider may use different contract technologies.

IV. Contract Policies

A. Contract Assertions

This section defines contract assertions and contract policies. An important feature is their representation in some neutral, programming language independent format. We apply the well-known WS-Policy standard for the following reasons: WS-Policy is supported by almost all Web services frameworks and is the standard formalism for enriching WSDL interfaces. With WS-PolicyAttachment [6], the principles for including policies into WSDL descriptions are specified.

WS-Policy defines the structure of so-called assertions and their compositions, but does not define the "content" of assertions. To represent preconditions, postconditions, and invariants, we need some adequate language. We decided to use the Object Constraint Language (OCL) due to its high degree of standardization and support by existing OCL libraries such as the Kent OCL Library, the Dresden OCL Toolkit, and the Open Source Library for OCL (OSLO).

To formally represent constraints with WS-Policy, we introduce so-called *contract assertions*. The XML schema as follows:

```
<xsd:schema ...>
  <xsd:element name = "ContractAssertion"/>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = "Precondition"
                   type = "xsd:string"
                   maxOccurs = "unbounded"/>
      <xsd:element name = "Postcondition"
                   type = "xsd:string"
                   maxOccurs = "unbounded"/>
      <xsd:element name = "Invariant"
                   type = "xsd:string"
                   maxOccurs = "unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "Name"
                   type = "xs:string"/>
    <xsd:attribute name = "Context"
                   type = "xs:anyURI"
                   use  = "required"/>
  </xsd:complexType>
</xsd:schema>
```

XML schema for contract assertions.

A `ContractAssertion` has two attributes: a mandatory *context* and an optional *name* for an identifier. The context attribute specifies the Web service to which the constraint applies. To be precise, the value of the context attribute is the name of the operation as specified in the `portType` section of the WSDL. In case of an invariant, the context attribute refers to the type defined in the `types` section.

The body of an contract assertion consists of a set OCL expressions. Depending on the surrounding element type the expression represents a precondition, a postcondition, or an invariant. The expressions may refer to the parameter and return values of an operation as well as to the attributes of a type.

B. OCL Expressions

OCL is a formal language for specifying particular aspects of an application system is a declarative manner. Typically, OCL is used in combination with the Unified Modeling Language (UML) to further constrain UML models. In OCL, "a constraint is a restriction on one or more values of (part of) an object-oriented model or system" [14]. In our context, OCL expressions will be used to specify constraints for Web services.

We use the following features of OCL in contract assertions:

- The basic types `Boolean`, `Integer`, `Real`, and `String`.
- Operations such as `and`, `or`, and `implies` for the `Boolean` type.
- Arithmetic (e.g., +, $\star$) and relational operators (e.g., =, <) for the types `Integer` and `Real`.
- Operations such as `concat`, `size`, and `substring` for the `String` type.

In order to impose restrictions on collections of objects, OCL defines operations for collection types. Well-known operations are:

- `size()`: returns the number of elements in a collection to which the method applies.
- `count(object)`: returns the number of occurrences of `object` in a collection.
- `includes(object)`: yields true if `object` is an element in a collection.
- `forAll(expr)`: yields true if `expr` is true for all elements in the collection.
- `select(expr)`: returns a subcollection containing all objects for which `expr` is true.

These operations may be used to constrain admissible values for collections occurring in the service's WSDL.

Before we give some examples, we introduce the keywords `@pre` and `result`, which can be used in postconditions. To impose restrictions on the return value of a service, the latter keyword can be used. In a postcondition, the parameters may have different values at invocation and termination, respectively, of the service. To access the original value upon completion of the operation, the parameter must be equipped with the prefix `@pre`.

C. Examples

The first example considers the square root service from Section II, extended by a postcondition. The following XML fragment shows a formulation as a contract assertion:

```
<ContractAssertion context="SquareRootService">
  <Precondition>
    d >= 0
  </Precondition>
  <Postcondition>
    return >= 0
  </Postcondition>
</ContractAssertion>
```

Contract assertion for square root service.

The identifier `d` in the precondition refers to the parameter name of the service as specified in the WSDL.

The next example illustrates two features: *i*) the definition of an invariant and *ii*) the usage of a path notation to navigate to members and associated data values. Consider the type `CustomerData` with members name, first name and address. If address is represented by another complex data type with members such as street, zip and city, we can apply the path expression `customer.address.zip` to access the value of the zip attribute for a particular customer instance.

Whenever an instance of `CustomerData` is exchanged between service provider and consumer, consistency checks can be performed as shown in the following figure:

```
<ContractAssertion context="CustomerDataService">
  <Invariant>
    this.name.size() > 0
  </Invariant>
  <Invariant>
    this.age >= 0
  </Invariant>
  <Invariant>
    this.address.zip.size() >= 0
  </Invariant>
</ContractAssertion>
```

An invariant constraint.

To demonstrate the usage of constraints on collections we slightly extend the example. Instead of passing a single `customerData` instance, assume that the service now requires a collection of those instances. Further assume that the parameter name is `cds`. In order to state that the collection must contain at least one instance, we can apply the expression `cds->size() >= 1`. With the help of the `forAll` operator one can for instance impose the constraint that the zip attribute must have a certain value: `cds->forAll(age = 78120)`.

V. Code Contracts and WCF

Having defined the solution architecture in Section III and contract policies in the previous section, we will now instantiate our approach. This section investigates Code Contracts for WCF and the following section applies Contracts for Java to JAX Web services.

A. Exporting Contract Policies

In WCF, additional WS-Policy descriptions can be attached to a WSDL via a so-called custom binding. Such a binding uses the `PolicyExporter` mechanism also provided by WCF. To export a contract policy as described in Section III, a class derived from `BindingElement` must be implemented. The inherited method `ExportPolicy` contains the specific logic for creating contract policies. Details for defining custom bindings and applying the WCF exporter mechanism are described elsewhere (e.g., [11]) and hence are not elaborated here.

B. Creating Contract Assertions

Code Contracts expressions are mapped to corresponding contract assertions. Thereby we distinguish between the creation of *i*) the embedding context and *ii*) OCL expressions for preconditions, postconditions, and invariants.

In Code Contracts, a precondition (resp. postcondition) is specified by a `Contract.Requires` statement (resp. `Contract.Ensures`). Thus, for each `Requires` and `Ensures` statement contained in the Web service implementation, a corresponding element (i.e., `Precondition` or `Postcondition`) will be generated. The context attribute of the contract assertion is the Web service to which the constraint applies.

According to the Code Contracts programming model, a class invariant is realized by a method that is annotated with the attribute `ContractInvariantMethod`. For such a method, the element `Invariant` will be created; its context is the type that contains the method.

Let us now consider the mapping from Code Contracts expressions to corresponding ones of OCL. We first observe that Code Contracts expressions may not only be composed of standard operators (such as boolean, arithmetic and relational operators), but can also invoke pure methods, i.e., methods that are side-effect free and hence do not update any pre-existing state. While the standard operators can be mapped to OCL in a straightforward manner, user defined functions (e.g., prime number predicate) typically do not have counterparts in OCL and hence will not be translated to OCL. For a complete enumeration of available OCL functions see [7], [14].

Due to lack of space we cannot discuss details of the mapping. The following table focuses on selected features:

Code Contracts	OCL
`0 <= x && x <= 10`	`0 <= x and x <= 10`
`x != null`	`not x.isType (OclVoid)`
`Contract.OldValue(param)`	`@pre param`
`Contract.Result<T>()`	`return`
`Contract.ForAll (cds, cd => cd.age >= 0)`	`cds.forAll (age >= 0)`

In the first two examples `x` denotes a name of an operation parameter. They illustrate that there are minor differences regarding the concrete syntax of operators in both languages. The third example shows the construction how to access the value of a parameter at method invocation. While Code Contracts provide a `Result` method to impose restrictions on the return value of an operation, OCL introduces the keyword `return`. In the final example, `cds` represents a collection; the expressions impose restrictions which must be fulfilled by all instances contained in the collection.

C. Importing Contract Assertions

As shown in Figure 1, the role of the importer component is to construct contract-aware proxy objects. WCF comes with the tool `svcutil.exe` that takes a WSDL description and produces the classes for the proxy objects. Note that `svcutil.exe` does not process custom policies, which means that the proxy objects do not contain contract assertions.

WCF provides a mechanism for evaluating custom policies by creating a class that implements the `IPolicyImporterExtension` interface. In our approach, we create such a class that realizes the specific logic for parsing contract assertions and for generating corresponding Code Contracts assertions. As the standard proxy class is a partial class, the created Code Contracts assertions can be simply included by creating a new file.

VI. Contracts for Java for JAX Web services

In this section, we consider a contract technology for Java. These principles of this description can be carried over to other Java based contracts technologies.

A. Exporting Contract Policies

In Contracts for Java [3], the preconditions, postconditions, and invariants are expressed with the annotations `Requires`, `Ensures`, and `Invariant`, respectively. An example has been given in Section II.

The reflection API of Java SE allows the inspection of meta-data. In order to access the annotations of methods we apply these API functions. Given a method (which can be obtained by applying `getMethods()` on a class or an interface), one can invoke the method `getAnnotations()` to get its annotations. Such an annotation object represents the contract expression to be transformed into an OCL expression.

Before we consider in more detail this transformation, we discuss how to create and embed contract policies into WSDL descriptions. A Web services framework provides API functions for these tasks; these functions are not standardized, though. As a consequence, we need to apply the specific mechanisms provided by the underlying Web services frameworks.

Basically, the developer has to create a WS-Policy with the assigned assertions. To include the policy file into the service's WSDL, one can use the annotation `@Policy`, which takes the name of the WS-Policy file and embeds it into the WSDL. Other frameworks create an "empty" default policy, which can be afterwards replaced by the full policy file. During deployment, the updated policy will be embedded into the service's WSDL.

B. Creating Contracts Assertions

In Contracts for Java, the expressions contained in the `@requires`, `@ensures`, and `@invariant` annotations are either simple conditions (e.g., `d >= 0`) or complex terms with operators such as `&&` and `||`. As in Code Contracts, the expressions may refer to parameter values and may contain side-effect free methods with return type `boolean`. Similar to the mapping of Code Contracts expressions, these methods will not be mapped to contract assertions (see Section V-B).

The following table gives some hints how to map expressions from Contracts for Java to OCL.

Contracts for Java	OCL
`0 <= x && x <= 10`	`0 <= x and x <= 10`
`x != null`	`not x.isType (OclVoid)`
`Contract.OldValue(param)`	`old(param)`
`Contract.Result<T>()`	`result`

Note that Contracts for Java currently does not provide special support for collections (such as a `ForAll` operator). Thus, a special predicate needs to be defined by the "contract developer".

C. Importing Contract Assertions

To obtain the (standard) proxy objects, tools such as `WSDL2Java` are provided by Java Web services frameworks. Given a WSDL file, such a tool generates Java classes for the proxy objects. In order to bring the contract constraints to the proxy class, we apply the following strategy:

1) Import of the contract policy contained in the WSDL.
2) Enhancement of the proxy classes by Contracts for Java expressions obtained from the contract policy.

There is no standardized API to perform these tasks. However, Java based Web services infrastructures have their own mechanisms. A well-known approach for accessing the assertions contained in a WS-Policy is the usage of specific importer functionality. To achieve this, one can implement and register a customized policy importer, which in our case generates `@requires`, `@ensures`, and `@invariant` annotations as required for the contract assertions contained in the WS-Policy.

The second steps interleaves the generated expressions with the standard proxy classes. A minimal invasive approach is as follows: Instead of directly enhancing the methods in the proxy class, we create a new interface which contains the required Contracts for Java expressions. The proxy objects must only slightly be extended by adding an "implements" relationship to the interface created. This extension can be easily achieved during a simple post-processing activity after `WSDL2Java` has been called.

VII. Limitations and Open Issues

We have already mentioned that contract languages are more expressive that OCL. They in particular allow the usage of user-defined predicates implemented, e.g., in Java or C#. As OCL it not a full-fledged programming language, not every predicate can be mapped to OCL. In other words, only a subset of the constraints will be available at interface level. At first sight, this seems to be a significant limitation. However, the role of preconditions and postconditions is usually restricted to perform (simple) plausibility checks on parameter and return values. OCL has been designed in this direction and hence supports such kinds of functions.

When specifying contracts for a class, one may impose constraints on private members, which are not visible at interface level. As a consequence, it is not helpful to map these constraints to contract policies for WSDL. Thus, the generated contract policies should only impose constraints which are meaningful to service consumers. To be precise, the generated contract assertions should only constrain the parameter and return values of Web services as well as the public members of complex data types contained in the `types` section of a WSDL.

Although WS-Policy [5] and WS-PolicyAttachment [6] are widely used standards, there is no common API to export and import WS-Policy descriptions. As mentioned before, Web services infrastructures have their specific mechanisms and interfaces how to attach and access policies. Thus, the solutions presented in this paper must be adapted if another Web services framework should be used. For instance, the exporter and importer classes for processing contract policies must be derived from different interfaces; also the deployment of these classes must be adapted.

Finally, we observe that the exception handling must be changed, if contract policies are used. This is due to the fact that the contract runtime environment has the responsibility to check the constraints. If, e.g., a precondition is violated, an exception defined by the contract framework will be raised, that contains a description of the violation (e.g., that the value of a particular parameter is invalid). This must be respected by the client developer.

VIII. Conclusion and Future Work

In this work, we have demonstrated how contract technologies designed for programming languages can be leveraged for the interface creation for distributed services. Constraints imposed on the implementation will now be part of the WSDL interface and hence visible to the Web service consumer. This approach is a further step to improve the quality of distributed software components. This concept is in particular useful when applying the *code-first* approach for developing Web services, since additional properties of the service implementation (e.g., preconditions) will be automatically mapped to contract policies.

The developer of a Web service client application explicitly sees important constraints imposed on the service implementation and hence can consider this information. In addition, constraint violations can be detected already on client side, thus reducing network traffic and server consumption.

Due to the usage of the standardized, “neutral” language OCL for expressing constraints, the interoperability between different contract technologies and Web services infrastructures is given. For instance, preconditions encoded on server side with Contracts for Java will be translated to corresponding Code Contracts statements in a .NET consumer application.

As the contracts can be generated automatically, no additional effort is required for the service developer. As shown in our proof of concept, current tool chains can be enhanced such that the creation of the contract polices are completely transparent for the developer. Similar tool support is possible for service consumer side.

In this paper, we have focused on Contracts for Java as a contracts technology for Java. However, there are other well-known alternative technologies. A closer look to these approaches and their usage for generating contract policies is part of future work.

Acknowledgments

I would like to thank Ahmed Al-Moayed and the anonymous reviewers for giving helpful comments on an earlier version. This work has been partly supported by the German Ministry of Education and Research (BMBF) under research contract 17N0709.

References

[1] B. Meyer, “Applying “Design by Contract”,” *Computer*, vol. 25, pp. 40–51, October 1992.

[2] Microsoft Corporation, “Code contracts user manual,” http://research.microsoft.com/en-us/projects/contracts/userdoc.pdf, last access on 08/15/2011.

[3] N. M. Le, “Contracts for java: A practical framework for contract programming,” http://code.google.com/p/cofoja/, last access on 08/15/2011.

[4] Java Modeling Language. http://www.jmlspecs.org/, last access on 08/15/2011.

[5] Web Services Policy 1.5 - Framework. http://www.w3.org/TR/ws-policy/, last access on 08/15/2011.

[6] Web Services Policy 1.5 - Attachment. http://www.w3.org/TR/ws-policy-attach/, last access on 08/15/2011.

[7] OMG, “Object constraint language specification, version 2.2,” http://www.omg.org/spec/OCL/2.2, last access on 08/15/2011.

[8] A. H. Anderson, “Domain-independent, composable web services policy assertions,” in *POLICY ’06: Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 149–152.

[9] WS-SecurityPolicy 1.3. http://docs.oasis-open.org/ws-sx/wssecuritypolicy/v1.3, last access on 08/15/2011.

[10] A. Erradi, V. Tosic, and P. Maheshwari, “MASC - .NET-based middleware for adaptive composite web services,” in *IEEE International Conference on Web Services (ICWS’07)*. IEEE Computer Society, 2007.

[11] B. Hollunder, “Code contracts for windows communication foundation (WCF),” in *Proceedings of the Second International Conferences on Advanced Service Computing (Service Computation 2010)*. Xpert Publishing Services, 2010.

[12] A. Goncalves, *Beginning Java EE 6 Platform with GlassFish 3*. Apress, 2009.

[13] D. Jayasinghe and A. Afkham, *Apache Axis2 Web Services*. Packt Publishing, 2011.

[14] J. Warmer and A. Kleppe, *The Object Constraint Language: Getting Your Models Ready for MDA*. Addison Wesley, 2003.

User-to-User Delegation in a Federated Identity Environment

HongQian Karen Lu

Gemalto
Austin, Texas, USA
Karen.lu@gemalto.com

Abstract - **Delegation protocols over the Web are mostly used for user-to-machine and machine-to-machine delegations. As more organizations operate in a federated identity environment, user-to-user delegation also becomes a required functionality. User-to-machine or machine-to-machine delegation methods cannot directly apply to user-to-user delegation because human cannot effectively process protocol messages. This paper proposes a new method that allows user-to-user delegations in a federated identity environment. The identity provider (IdP) acts as the delegation authority that manages delegations. Service providers (SPs) in the same environment can use this delegation service, instead of managing delegations individually. The service includes delegation assignment, invocation, and revocation. The method allows service providers to exercise access controls and to decide if the delegator has the right to delegate and if the delegatee should be authorized to perform the requested services. This method is applicable to any access control models.**

Keywords - access control, delegation, federated identity, security.

I. INTRODUCTION

A privilege is the right to perform a certain action to a specific resource or resources; for example, to read (action) a file (resource). Delegation is a process of an identified entity, called a delegator, giving some of the delegator's privileges to another identified entity, called a delegatee. The delegatee receives the privileges to act on behalf of the delegator at a service provider [1][2].

The delegation can be user-to-user (or called person-to-person), user-to-machine, or machine-to-machine. The person-to-person delegation happens often in the physical world. In the digitalized world, a person (a user who uses a computer, an application, or a system) has certain privileges or access rights at a service provider (SP). She may want to give some of her privileges to another user under certain conditions. For example, Alice delegates some of her responsibilities at an SP to Bob while she is out of her office. When a user access services at a SP and the SP needs to access the user's resources at another SP on the user's behalf, the user can authorize a delegation to the first SP, which is a user-to-machine delegation. The machine-to-machine delegation happens similarly among service providers.

The continued increase of online collaborations among organizations and service providers has brought the need of *federated identity* [3]. A federated identity environment consists of an identity provider (IdP) and one or more service providers (SP). The IdP manages user identities and authenticate users. The SPs provide web services and trust the IdP‘s assertions about the users. Typically, IdP and SPs are different entities and in different domains. This construct, among other things, enables Single Sign-On, which allows a user to use a single set of credentials to login to different SPs through the IdP and login once under certain conditions. This is convenient for both users and SPs, and potentially can provide stronger authentication and, hence, better security as well. Reference [3] provides a good overview about the need and use cases of federated identity, and roles of IdP and SP.

Most research on delegation in a federated identity environment focuses on user-to-machine or machine-to-machine delegations [4]. As the Web becomes the ubiquitous computer, and more organizations, government, and businesses operate in and depend on federated identity environments, user-to-user delegation over the web in such an environment becomes a required functionality. User-to-machine or machine-to-machine delegation methods cannot directly apply to user-to-user delegation because human cannot effectively process delegation protocol messages that may require complex computational and cryptographic operations.

User-to-user delegation has been studied extensively in role-based access control (RBAC) systems [5], and is typically used with a specific SP. In this case, the SP manages its delegation service, which is not an easy task. Furthermore, the delegation should work with different access control models in addition to RBAC.

We propose a new delegation method to address the above issues. The method supports user-to-user delegation service in a federated identity environment. The delegation allows a user to delegate some of her privileges at an SP to another user. The IdP acts as the delegation authority that manages delegations. The SPs use this delegation service, instead of managing delegations individually. The delegation service includes delegation assignment, invocation, and revocation. The SPs need to ensure that delegations are compliant with their access control policies. To facilitate this, the delegation method provides opportunities for SPs to consult their access control engines in order to decide if the delegator has the right to delegate

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-152-6

and if the delegatee should be authorized to perform the requested services. Therefore the method is not tied to any particular access control models.

The interactions between SPs and an IdP follow the SAML 2.0 [6] and XACML [7] standards. We use standard syntax, assertions, protocols, and bindings as much as possible and extend them only as needed.

The rest of the paper is organized as follows. Section II provides some background information and outlines the related work. Section III presents the new user-to-user delegation method in a federated identity environment. Section IV describes the protocols used to support the delegation schemes. Section V discusses issues related to security and implementation. Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

We follow the delegation terminology defined in the reference [4]. A *privilege* is the right to access specific resources or to perform certain tasks. A user may have a number of such privileges. *Delegation* is an act of (temporarily or permanently) transferring privileges from one entity to another. A *delegator* is an entity that transfers (delegates) all or a subset of its privileges to a delegatee. A *delegatee* is the entity that receives the delegator’s privileges in order to use them on the delegator’s behalf. A *delegation assertion* is an assertion of the correctness and authority for a delegation, issued by a delegation authority to a delegatee. A *delegation authority* is an entity that controls delegation and issues delegation assertions.

A. Access Control and XACML

Access controls are security mechanisms that control how subjects (users, applications, and systems) access and interact with objects (resources, other applications, and systems). Access control includes identification, authentication, authorization, and accountability. There are three main types of access control models: discretionary, mandatory, and role-based. All organizations must have access control policies and implementations to protect their resources and systems from unauthorized access.

Delegation is a mechanism of transferring access privileges from one subject to another. Such a privilege transfer must be authorized and not violate organization/system’s access control policy. Therefore, the access control policy should include delegations, and the delegation mechanism should be associated with the access control engine. Research on delegation in access control models have built on and supported this concept. However, recent research on delegation related to the federated identity and machine-to-machine delegations have focused on mechanisms and semantics but failed to address the link between the delegation and access control [2][4][8].

The XACML (eXtensible Access Control Markup Language) is a standard for specifying and communicating access control policies across computer systems (internal or external to an organization) [7]. The current version is XACML 2.0. The XACML 3.0 is working in progress, which includes the concept and process of delegation. The XACML delegation deals with creation of new policies and tracing back trusted policies. We can use the syntax of the XACML 3.0 delegation to support our work.

B. User-to-User Delegation

Delegation in Role-Based Access Control (RBAC) has been studies extensively [5]. The RBAC system manages the delegation based on the access control policies. In doing so, it must answer the following two questions: 1. Is a user (delegator) authorized to delegate a role, privilege, or permission that is available to him? 2. Can a role, privilege, or permission be delegated to a user (delegatee)?

In the context of delegation service in a federated identity environment, service providers manage their own access controls and, hence, the permission to delegate. SP must find answers to the above questions when a user delegates, when a user invokes a delegation, and when a user's privileges have changed. This applies to any access control model, not just RBAC.

Peeters *et al.* [2] described procedures of the delegation, including mandate issuance, acceptance, revocation, and invocation. Furthermore, the paper outlined advanced delegations: transferable delegation and corporate delegation. The paper stays at a pure conceptual level and not at the web application level. The title including “identity federation” is somewhat misleading as the approach has no link to any federated identity method.

C. User-to-Machine Delegation

The Shibboleth System is a SAML 2.0 based, open source software package for web Single Sign-On across or within organization boundaries [9]. The Shibboleth has a solution to the proxy authentication problem: how to authenticate a service to which a user may have authenticated to, and who wishes to invoke another service on the user's behalf [8]? The method uses two Single Sign-On’s through the same IdP. The delegation assertion is enabled by the first authentication statement and is built into the second authentication statement. The IdP issues and signs the delegation assertion.

Alrodhan and Mitchell [4] proposed a delegation framework for Liberty Alliance Project. The method extends the attribute statement in the SAML assertion to form a delegation assertion. The IdP issues and signs the delegation assertion with the user (delegator, privilege owner) consent. The Single Sign-On Profiles described in the Liberty ID-FF 1.2 specification provides the base for this delegation framework. This work is similar to Shibboleth's delegation in the sense that the delegatee (first SP), through the user agent, gets a delegation assertion from the IdP and then presents it to the target (second SP). The differences are in profiles, assertions, and so forth.

OAuth is an open protocol that enables a website to access protected resources from another website on the resource owner's behalf, without requiring the resource owner to disclose his login credentials [10]. As such, OAuth provides a protocol for user-to-machine delegation. A growing number of companies support OAuth, including Twitter, Google, Netflix, Yahoo!, and Facebook.

The "SAML V2.0 Condition for Delegation Restriction" specifies the expression of delegation information through a SAML Condition extension to address the use cases that a single logical transaction involves one or more intermediate entities (clients or servers) [11]. The SP must evaluate delegates in the condition and should only accept the assertion if it wishes to accept the condition.

D. *Resource Sharing*

The Liberty Alliance Identity Web Service Framework (ID-WSF) People Service (PS) [12] is a web service that allows users to track and manage people they know online, and allows other web services to query and manage the people list of their users.

The People Service enables "cross-user" interactions that involve more than one user for an online activity. For example, Alice wants to share her photos with Bob at her photo website at which Bob does not have an account. The photo website uses Alice's People Service and Bob's identity provider to identity Bob and lets him to access photos specified by Alice. Such resource sharing may be used for user-to-user delegation in limited situations, but is not designed for such a purpose. It implicitly assumes the discretionary access control (DAC). If Bob also has an account with the SP, the People Service is unnecessary.

III. USER-TO-USER DELEGATION

We propose a new method that supports user-to-user delegation service in a federated identity environment. The delegation allows a user to delegate some of his privileges at a service provider (SP) to another user. The delegation service includes delegation assignment, invocation, and revocation.

In a federated identity environment, service providers trust the identity provider (IdP) to manage user identities and authenticate users [3]. In such an environment, IdP can, in addition, act as the delegation authority that manages user-to-user delegations. The delegator assigns delegations at IdP. The delegatee is to perform the delegated tasks at the specified service provider (SP). The SP obtains delegation assertions from the IdP. Delegations can be revoked either by the delegator or by the SP.

Why do we need a delegation authority? Each individual SP can certainly provide the delegation service by itself without needing a delegation authority. However, tracking and managing delegations is not a trivial task. A valid alternative for service providers is to use a trusted delegation authority [2][4]. From a user perspective, a delegator may want to delegate at more than one service provider. Going to each service provider one by one is not convenient, at least. A delegation authority can solve this problem by providing a common portal for the delegation service.

The service providers' access control models play an important role in the design of this delegation method. Service providers need to ensure that delegations are compliant with their access control policies. For this purpose, the delegation method allows SPs to exercise access controls throughout delegations' life cycles. These mechanisms are independent of SPs' access control models.

A. *Delegation Life Cycle*

When a delegator delegates to a delegatee, the IdP creates a delegation record, or simply called a delegation. A delegation life cycle starts with the creation and ends with the deletion. The following figure illustrates the life cycle.

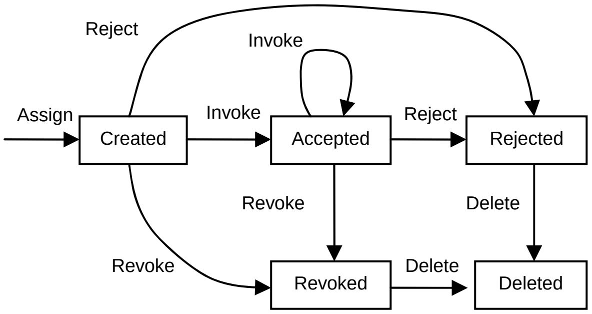


Figure 1. Delegation life cycle.

A delegation record includes the delegator, the service provider, the delegatee, the resources that the delegatee is to access, the actions that the delegatee can do after obtaining the resource, and other things, such as assignment date and time, valid period, delegator's signature, and so on. When the delegator assigns a delegation, the IdP creates a delegation record; the delegation is in the created state. The invocation of the delegation by the delegatee transfers the delegation into the accepted state. Other states illustrated in the figure are self-explainable.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-152-6

B. Delegation Schemes

The main delegation schemes include assignment, invocation, and revocation.

Assignment

A delegator wants to delegate to a delegatee some tasks to be performed at an SP. The IdP does not know what privileges that the delegator can delegate. The IdP's role is to manage the delegation and to tell the SP that the delegator indeed has delegated certain privileges to the delegatee. The SP needs to make sure that the delegator has these privileges and can delegate them, and the delegatee is authorized to perform the tasks.

If the delegator knows exactly what he can delegate, the assignment becomes simple. The delegator specifies the SP, delegatee, and privileges that he wants to delegate. The IdP creates a delegation record. In many cases, however, the delegator may know what he wants to delegate to a delegatee but is not sure if he can. Then the assignment is more complex. The IdP needs to ask the SP if it can authorize a specified delegation request. The IdP does this by, for example, making an `XACMLAuthzDecisionQuery`, which is specified in the XACML SAML profile. If SP responds with a success, IdP creates a delegation. Otherwise, IdP asks the delegator to make a modification and repeat the process.

In general, a delegator may not know if or what he can delegate to a particular delegatee. In this case, he specifies the SP and the delegatee. The IdP asks the SP about privileges that the delegator can delegate to the delegatee. The SP responds with a list, which may be empty. The IdP asks the delegator to make a selection. When needed, the IdP asks the SP if such delegation can be authorized. If SP responds with a success, IdP creates a delegation record.

Figure 2 illustrates the delegation assignment workflow of the above general case, which includes the following steps. We can easily adopt this workflow to simpler cases.

1. The delegator **A** authenticates to the IdP.
2. **A** selects the SP that he wants the delegatee **B** to access.
3. The IdP finds from the SP the privileges that **A** can delegate to **B**.
4. The IdP presents a list containing those privileges (resources, actions) to **A**.
5. **A** selects privileges to delegate to **B** from the list and other constraints, such as valid time period.
6. The IdP creates a delegation record. (Optionally, IdP asks the SP if such delegation can be authorized before creating a delegation.)
7. The IdP may ask **A** to digitally sign the delegation for non-repudiation.
8. **A** signs the delegation if required.
9. **A** or the IdP informs **B** about the delegation.

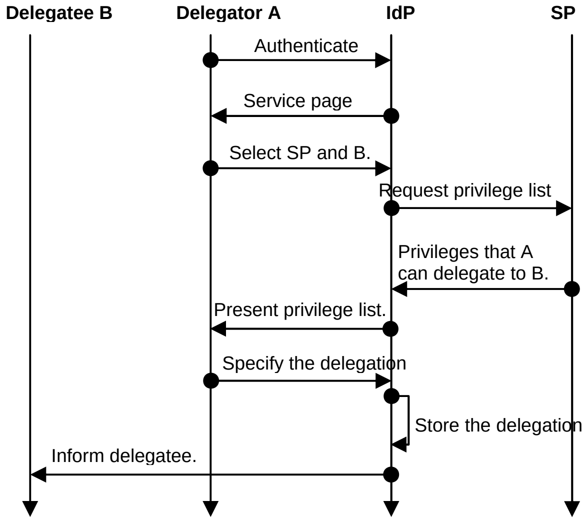


Figure 2. Delegation assignment.

The SP checks with its access control engine to decide what privileges that **A** can delegate to **B** and presents the privilege list, if exist, to IdP. The access control engine makes decisions according to its access control policies.

In practice, a delegator may need an approval from her manager or some other entities in order to delegate. The access control system may not have an automated mechanism for doing so. Then the approval is a physical process. Addressing such issues is outside the scope of this paper.

Invocation

When the delegatee requests to perform a delegated task at the SP, he invokes a delegation. Figure 3 illustrates this process, which consists of the following steps:

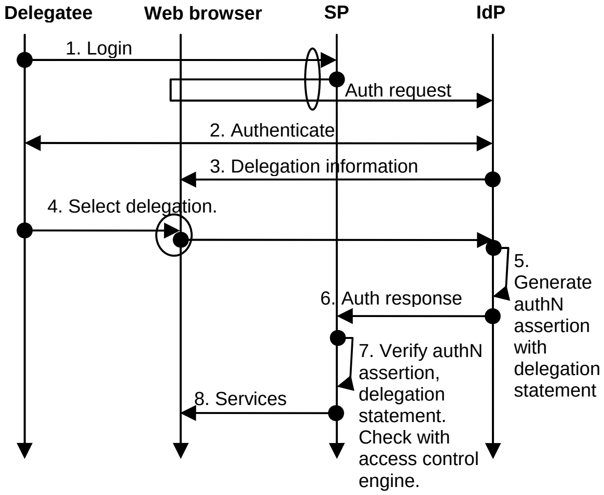

Delegatee
Web browser
SP
IdP
1. Login
Auth request
2. Authenticate
3. Delegation information
4. Select delegation.
5. Generate authN assertion with delegation statement
6. Auth response
7. Verify authN assertion, delegation statement. Check with access control engine.
8. Services

Figure 3. Invocation.

1. The delegatee logs in to the SP, which redirects the authentication to the IdP.
2. The IdP authenticates the delegatee.
3. The IdP finds and presents delegation(s) at the SP to the delegatee.
4. The delegatee selects one or more delegations.
5. The IdP generates an authentication assertion for the delegatee with a delegation attribute statement specifying the delegation(s).
6. The IdP sends the authentication assertion to the SP.
7. The SP verifies the authentication assertion and the delegation statement. The SP consults with its access control engine for both the delegator and delegatee.
8. If all is well, The SP present services to let the delegatee to perform the delegated tasks.

The IdP generates an authentication assertion in response to the authentication request from the SP. The subject in the assertion is the delegatee. The assertion in addition includes an attribute statement about the delegation. The following code snippet illustrates an example in the form of an SAML 2.0 assertion.

```
<Assertion>
  <Issuer> … URI of the IdP … </Issuer>
  <ds:Signature> IdP’s signature </ds:Signature>
  <Subject> Information on delegatee </Subject>
  <Conditions>
  <AuthnStatement> // authentication statement
  <AttributeStatement>
    <Attribute Name=”Delegation”>
       <AttributeValue>
          <Delegator>
          <Delegatee>
          <Privilege> // one or more
             // description, services, resources,
actions, and so forth.
          </Privilege>
       </AttributeValue>
    </Attribute>
  </AttributeStatement>
</Assertion>
```

The service provider processes the delegation information. For example, SP verifies the following:

1. The delegation is in the valid period.
2. The service request is specified in the statement.
3. The requester is the delegatee specified in the statement.
4. The signature of the assertion is valid and the certificate is not revoked.
5. The delegator is authorized to perform the delegated privileged task.
6. The delegator is authorized to delegate the privileged task.
7. The delegatee is authorized to perform the delegated task.
8. Other optional constraints are met.

If any of the verification steps fails, the SP denies the requested service from the delegatee. The checking on authorizations is necessary because conditions may have changed since the last time that the SP queried the access control engine regarding the delegator and the delegatee when the IdP requested the privilege list in setting up the delegation. If any of the authorization checking fails, the SP revokes the delegation.

Revocation

Previous research suggested using certificate revocation mechanisms, Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP), to handle delegation revocation [2]. Both CRL and OCSP are known for their complexities of maintaining the list of revoked certificates. We propose a simpler delegation revocation approach that does not require maintaining a revocation list nor require a separate query on the delegation status.

As described earlier, IdP is the delegation authority that manages delegation records. SP gets the delegation statement from IdP as a part of an authentication assertion. The SPs should not accept delegations from anyone else. SP can store the delegations for audit purposes, but should not reuse them. The delegation assertion is always dynamically acquired. Therefore, SP does not need to check for the delegation status.

A delegation revocation can be initiated by the delegator or by the SP. After receiving a revocation request, IdP authenticates and verifies the request. If the request is authentic and verifiable, IdP removes the delegations involved from its delegation database. (IdP may keep the revoked delegations for auditing purpose.)

The delegation situation is very different from that of SSL certificate. Typically SP receives a SSL certificate from a third party. Before using it, SP would check to see if the certificate is still valid by consulting with CRL or using an OCSP service. With our delegation approach, SP obtains the delegation assertion from the delegation authority, IdP. Therefore, SP can verify the assertion and does not need to ask IdP for the validity of the assertion, and IdP does not need to provides a service for such purpose.

(a) Revocation by Delegator

The delegator can revoke a delegation at IdP. This involves the following steps:

1. The delegator **A** logs in to the IdP.
2. **A** revokes his delegation to the delegatee **B**.
3. IdP removes **A**’s delegation to **B** from its record.
4. **A** or IdP informs **B** about the revocation of the delegation.

(b) Revocation by Service Provider

When a user’s privilege**s** are reduced or removed, the service provider should find out if there are outstanding delegations relevant to this user. If so, SP needs to examine each of the delegations to see if they are still valid. For example, if the user was a delegator and he no longer has privilege to delegate the task, or if the user was a delegatee and he no longer has the privilege to perform the delegated task, then SP sends a revocation request to IdP to revoke the delegation. IdP then informs the delegator and the delegatee. Figure 4 illustrates the process. Section 4 will provide more details on the delegation query request and response, and delegation revocation request and response.

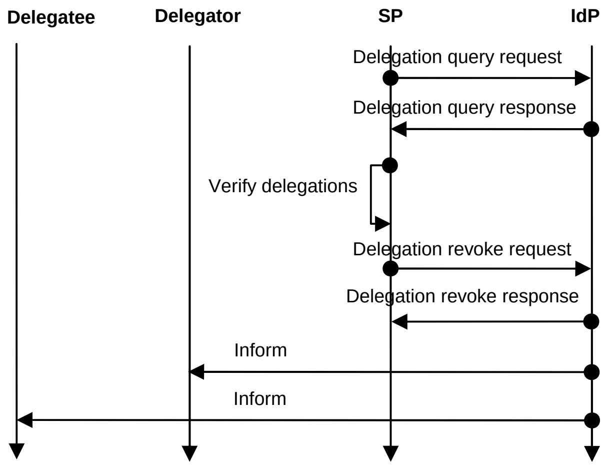


Figure 4. Revocation by the service provider.

(c) Cleanup by Identity Provider

IdP cleans up its delegation repository periodically. For delegations that have not been activated for a while or just for any delegations, IdP can make XACML authorization decision query to SP. If the response is negative, IdP can remove the delegation. This avoids using any SAML extension for delegation revocation.

C. Acceptance and Rejection

The delegatee can either accept or reject a delegation. The present scheme does not allow partial acceptance. The delegatee examines the delegation from the received information or the IdP provides a service for the delegatee to do so. The act of the delegatee requesting to perform the delegated task at the SP is a form of delegation acceptance. IdP may also provide a service for the delegatee to explicitly accept the delegation.

If the delegatee rejects the delegation, he can inform the delegator, who may either modify the delegation or revoke the delegation at IdP. The IdP may also provide a service for the delegatee to reject a delegation. Revocation of acceptance can be done the same way as rejection.

IV. PROTOCOLS

Delegation assignment, query, invocation, and revocation all require communications between SP and IdP. We use SAML 2.0 assertions [6] as the message exchange format and extend as needed. SAML protocols and bindings are used to transport the delegation messages.

A. Request and Response

SAML protocol is a request and response protocol. The requester sends a request, and the responder processes the request and sends a response.

B. Attribute Query

The SAML 2.0 attribute query `<AttributeQuery>` is used for querying attributes of a subject. We use it to query privileges that a delegator (subject) can delegate to a delegatee, and existing delegations for a delegator or delegatee. The response is an attribute assertion or query status.

Query Privileges

During the delegation assignment, IdP asks SP what privileges that the delegator can delegate to the delegatee. The XACML policy query `<XACMLPolicyQuery>` specified in the XACML SAML profile can serve this purpose. In response, the SP sends an XACML policy assertion that contains the requested information. The following code snippet illustrates the query.

```
<xacml-samlp:XACMLPolicyQuery>
  <saml:Issuer>
  <ds:Signature>
  <Attribute ID>
  <Attribute IssurInstant>
         … …
  <xacml-context:Request>
    <xacml:Attributes>
        <Attribute name=”user”>
          <AttributeValue>
             id or other attributes of delegator
          </AttributeValue>
        </Attribute>
        <Category
        name=”urn.oasis:names.tc:xacml:3.0:attribu
te-category:delegate”>
    </Attributes>
    <xacml:Attributes>
        <Attribute name=”user”>
          <AttributeValue>
            id or other attributes of delegatee
          </AttributeValue>
        </Attribute>
        <Category
name=”urn.oasis:names.tc:xacml:3.0:attribute-
category:delegated:urn:oasis:names:tc:xacml:3.0:su
bject-category:access-subject”>
    </Attributes>
        <xacml:Attributes>
        <Category
name=”urn.oasis:names.tc:xacml:3.0:attribute-
category:delegate”>
        <Category
name=”urn.oasis:names.tc:xacml:3.0:attribute-
category:delegated:urn:oasis:names:tc:xacml:3.0:su
bject-category:resource”>
        <Category
name=”urn.oasis:names.tc:xacml:3.0:attribute-
category:delegated:urn:oasis:names:tc:xacml:3.0:su
bject-category:action”>
    </Attributes>
  </Request>
  <Attribute name=”ReturnPolicyIdList”>
    <AttributeValue>true</AttributeValue>
  </Attribute>
        … …
</XACMLPolicyQuery>
```

In response, SP sends a <samlp:Response>, which contains an XACMLPolicy assertion that has a statement of the type xacml-saml:XACMLPolicyStatementType. This statement contains policies that the query requested.

Query Delegations

When a user’s privileges have been removed or reduced, the SP should examine all outstanding delegations associated with this user, either as a delegator or as a delegatee. For this purpose, the SP sends a SAML query request <AttributeQuery> to the IdP and the IdP responds with an attribute assertion containing relevant delegation statements, if they exist. The following code snippet illustrates the query.

```
<samlp:AttributeQuery>
  <saml:Issuer>
  <ds:Signature>
  <Attribute ID>
  <Attribute IssuerInstant>
        … …
  <Subject>
    <NamdID id=… delegator or delegatee’s id…>
                  … …
  </Subject>
  <Attribute name=”Delegation”>
        … …
</AttributeQuery>
```

The IdP responds with an assertion containing one or more attribute statements about the delegations.

C. *Authentication Request*

The authentication request is the standard SAML 2.0 <AuthnRequest>. When sending an authentication request, the SP does not know anything about the delegation. The delegatee selects the delegation at the IdP during authentication.

D. *Delegation Revocation Request*

When access privileges of a user have changed and existing delegations are no longer valid, the SP sends a delegation revocation request to the IdP to revoke relevant delegations. While neither SAML 2.0 nor XACML 2.0/3.0 specified revocation, we can follow the SAML syntax to define it. The delegation revocation request and response are similar to authentication request and response, in which the SP sends a request to the IdP; the IdP fulfills the request and sends an assertion in response. The request can take the following form.

```
<DelegationRevokeRequest>
  <Issuer>
  <ds:Signature>
        … …
  <Subject>
  <Attribute name=”Delegation”>
    <Delegator>
    <Delegatee>
    <Resource>
        … …
  </Attribute>
        … …
</DelegationRevokeRequest>
```

The elements in <Delegation> are optional. The request can have the following rules:

1. If no <Resource> is specified, SP requests to revoke all delegations associated with <Delegator>, <Delegatee>, or both.
2. If none of <Delegator> and <Delegatee> exists, SP requests to revoke all delegations associated with the subject and <Resource>.
3. If there is <Delegator> but no <Delegatee>, SP requests to revoke all delegations that the subject is a delegator and delegated <Resource> to any delegatee.
4. If there is <Delegatee> but no <Delegator>, SP requests to revoke all delegations that the subject is a delegatee and was delegated for <Resource> by any delegators.
5. If there are both <Delegator> and <Delegatee>, SP requests to revoke all delegations that associated to <Delegator>, <Delegatee>, and <Resource>. The subject can be a delegator or delegatee.

The response from the IdP contains the status of the revocation.

E. *Bindings*

A transport binding is a mapping from SAML messages to a communication protocol. The delegation statement comes as a part of an authentication assertion. Therefore, the delegation takes on whatever binding that the authentication process uses. For example, the authentication can use SAML 2.0 Web browser SSO profile [13]. The corresponding bindings include HTTP redirect, HTTP POST, and artifact bindings.

V. DISCUSSIONS

This section discusses some issues related to security and implementations. Other technical details for providing delegation services are dependent on the specifics of the environment, access control policies, security level, and so forth, which is outside the scope of this paper.

The proposed user-to-user delegation scheme can use existing federated identity frameworks and protocols, such as SAML 2.0 and XACML, as its foundation. The established trust relationships in a federated identity environment enable the SPs to trust and use the IdP as the delegation authority.

When deploying the delegation service, securing the communications between IdP and SPs is important, especially for services involving high value or high security transactions. At the web application level, this can be achieved by using HTTPS and TLS/SSL with mutual authentication and strong cipher suites. It allows each party to know for sure whom it is talking to, and ensures the integrity and confidentiality of the communications.

Digitally signing all delegation statements, queries, requests, and responses is also important. These signatures provide authenticity and non-repudiation. The SPs should not reuse delegation statements because the situation may have changed since a delegation was issued.

The delegation statement in the authentication assertion provided by the IdP is not an authorization of delegation. Instead the IdP vouches that the delegator indeed has delegated some tasks to the delegatee. It is the service provider's responsibility to consult its access control engine to decide if the delegatee should be authorized to receive the requested services, and to record the transactions.

This delegation scheme requires that the delegatee has an account at the IdP, because the IdP needs to be able to identify and authenticate the delegatee. The service providers' access control policies dictate whether the delegatee needs an account at their websites. For example, if the SP has a mandatory access control policy, the delegatee needs an account at the SP because the access control is managed by the SP's system. For another example, if the SP has a discretionary access control policy, that is, it lets the user to decide permissions regarding to *his* resources, such as data, at the SP. Then the SP may not need to know the identity of the delegatee.

VI. CONCLUSIONS

This paper proposes a new delegation method that enables user-to-user delegations in a federated identity environment. This method allows service providers (SPs) to use the delegation service, instead of managing delegations individually. The service providers can exercise access controls and decide if the delegator has the right to delegate and if the delegatee should be authorized to perform the requested services. This method is applicable to any access control models because service providers control the access to their resources.

ACKNOWLEDGEMENT

The author would like to thank Dr. Ksheerabdhi Krishna and Mr. Kapil Sachdeva for their help to this work.

REFERENCES

[1] X. Huysmans and B. Van Alsenoy, editors, "Identity management for eGovernment, Annex I. Glossary of terms (v1.07), IDEM project, 2007, https://projects.ibbt.be/idem/uploads/media/2007-12-27.idem.glossary.v1.07.pdf [accessed: July 18, 2011].

[2] R. Peeters, *et al*. "Cross-context delegation through identity federation," Proc. of SIG on Biometrics and Electronic Signature, 2008. http://www.cosic.esat.kuleuven.be/publications/article-1156.pdf [accessed: July 18, 2011].

[3] OASIS, "Security assertion markup language (SAML) v2.0 technical overview," editors: N. Ragouzis *et al*., committee draft 02, March 25, 2008, http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf [accessed: July 18, 2011].

[4] W. Alrodhan and C. Mitchell, "A delegation framework for liberty," Proc. of the 3rd Conference on Advances in Computer Security and Forensics, 10-11 July 2008, Liverpool, UK, http://www.isg.rhul.ac.uk/cjm/adffl.pdf [accessed: July 18, 2011].

[5] J. Crampton and H. Khambhammettu, "Delegation in role-based access control," Lecture Notes in Computer Science, volume 4189, 2006, pp. 174-191.

[6] OASIS, "Assertions and protocols for the OASIS security assertion markup language (SAML)," v2.0, http://saml.xml.org/saml-specifications [accessed: July 18, 2011].

[7] OASIS, "eXtensible access control markup language (XACML)," http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml [accessed: July 18, 2011].

[8] Internet 2 – Shib-uPortal, https://spaces.internet2.edu/display/ShibuPortal/Home [accessed: July 18, 2011].

[9] Shibboleth, http://shibboleth.internet2.edu/ [accessed: July 18, 2011].

[10] OAuth, http://oauth.net/ [accessed: July 18, 2011].

[11] OASIS, "SAML V2.0 Condition for delegation restriction," Committee draft 01, 10 March 2009, http://wiki.oasis-open.org/security/SAML2DelegationCondition [accessed: July 18, 2011].

[12] Project Liberty, "Liberty ID-WSF people service specification," v 1.0, http://projectliberty.org/liberty/content/download/890/6246/file/liberty-idwsf-people-service-v1.0.pdf [accessed: July 18, 2011].

[13] OASIS, "Profiles for the OASIS security assertion markup language (SAML)," v2.0, http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf [accessed: July 18, 2011].

A Pragmatic Online Authentication Framework using Smart Cards

H. Karen Lu, Asad Ali, Kapil Sachdeva[1]
Gemalto
Austin, Texas, USA
{Karen.lu, asad.ali}@gemalto.com

Ksheerabdhi Krishna
Gemalto
La Ciotat, France
Ksheerabdhi.krishna@gemalto.com

Abstract - **Like most security systems, designing a secure two-factor online authentication framework is hard, but designing one that is also intuitive to use and easy to deploy is even harder. While a secure, but overly complex framework may offer little security in the end since it never gets used, an overly simplistic one that focuses merely on usability may gain initial acceptance but will inevitably lead to data breaches. To address this design paradox, we present a new online authentication framework that provides security, usability, and ease of deployment. This framework combines the proven hardware security of smart cards and the universal ease of web access through browsers, without imposing the deployment and usability complexities generally associated with conventional smart card systems. The resulting authentication solution is applicable to existing smart cards already deployed, intuitive for users, and convenient for service provides to both develop and maintain.**

Keywords-Authentication; security; smart cards; usability.

I. INTRODUCTION

Internet has undoubtedly been a phenomenal success, dominating every facet of our professional and social life. However, this success has partly come at the expense of a continuous barrage of security attacks against both users and service providers. Attackers employ various mechanisms to steal user's credentials. Some use social engineering to lure naïve users into revealing their credentials [1], while others leverage network security flaws and web application vulnerabilities to attack web servers and their databases [2]. These attacks compromise confidential user data. Some of this data can actually be user authentication credentials that enable attackers to impersonate users and gain subsequent access to additional user data and services. This is generally referred to as identity theft. Such theft is possible partially because a vast majority of online service providers still rely on username and password, a weak single-factor authentication method. Furthermore, since users tend to use the same password on multiple service providers [3], it amplifies the potential damage resulting from a stolen credential.

The weakness of password based authentication solutions can be addressed by using an authentication method that relies on multiple factors for verifying a user's identity. For example, in addition to password, the what-you-know factor, the authentication method may also require a what-you-have factor in the form of a separate physical token, or even a what-you-are factor in the form of biometric information. While there is some social skepticism around the use of biometric information, the use of dedicated physical tokens to provide a second authentication factor that compliments passwords is gradually gaining acceptance with service providers dealing with high value transactions [4]. In general however, we still see a lot of not-so-secure systems in use. One reason for this could be the inertia of status quo; it is always hard to change an existing framework. Another reason is what we call economies of convenience. This notion is somewhat analogous to the economies of scale, a microeconomic term that refers to the cost advantages that a business obtains due to expansion. Similarly, there is also a cost advantage to having systems that are extremely convenient to use, even if they are not as secure. Enterprises can then develop risk models of dealing with data breaches, when they happen. As for the average end-users, they generally turn a blind eye to security vulnerabilities as long as the systems they use are convenient, and security threats not imminent.

However, a continued increase in the intensity and frequency of cyber attacks is beginning to challenge these well established economies of convenience. Enterprises will eventually mandate stronger security measures once it makes better economic sense for them to lower the cumulative cost of data breaches by reducing the risk instead of managing this risk with their current models. We can reach this watershed moment either through an exponential increase in the number of data breaches, or by designing security systems that are more convenient to develop, deploy, use, and manage. It is the intent of this paper to propose a solution for the later.

The rest of the paper is organized as follows. Section II describes why smart cards are excellent candidates for use as authentication tokens. Section III describes the existing smart card infrastructure and explains how it hinders wide spread adoption of smart cards. Section IV introduces SConnect technology that addresses the issues identified in Section III. Section V describes a two-factor online authentication solution based on SConnect and

[1] This work was completed while Mr. Sachdeva was with Gemalto. Mr. Sachdeva is now working with HID.

ISBN: 978-1-61208-152-6

Section VI offers security and usability analysis of this solution. We conclude with Section VII.

II. AUTHENTICATION TOKENS

Physical tokens for multi-factor online authentication generally use one of the two common authentication techniques; One-Time-Password (OTP), or X.509 certificate based challenge and response. In both cases the hardware processor of the token uses private keys to perform cryptographic computations for generating a "credential" that is unique for each authentication attempt, and therefore can neither be stolen from the web server, nor replayed by an attacker. Since the token stores the private cryptographic keys, the strength of such an authentication method is a function of the token's hardware security.

Smart cards are excellent candidates for these physical tokens. They are tamper resistant, portable, and secure microprocessor devices that have been widely used in a variety of applications related to both physical and logical security. The smart card does not usually have its own power supply, yet it operates as a very small computer with an embedded operating system (OS) that controls application execution, access restrictions and communication with the outside world. However, unlike the mainstream personal computers, smart cards offer much greater hardware security. It is extremely difficult to compromise data stored inside the smart cards. This is because smart cards are designed with a heavy focus on security from the ground up, and this focus is maintained throughout their lifecycle. As such, smart cards can withstand attacks based on physical probing, logical probing, side channel threats, fault induction and software debugger probing [5]. A more detailed discussion of techniques for preventing such attacks is outside the scope of this paper.

Suffice to say that smart cards can serve as excellent tokens of two-factor authentication. However, despite their hardware advantage, smart cards are yet to garner widespread adoption outside their controlled niche markets. One reason for this lackluster acceptance is the complexity of deploying smart card based solutions, and the inconvenience of using them. To address these problems this paper introduces a new two-factor online authentication framework. It supports an X.509 certificate-based challenge-response model of authentication using smart cards, and utilizes a unique communication model that allows seamless access to smart card functionality directly from web applications. This approach facilitates easy adoption by end users as well as service providers.

III. CURRENT SMART CARD FRAMEWORK

In order to appreciate the value of the new method, we first have to consider how smart cards are currently used for online authentication. This use is somewhat restricted to environments where it is viable to create and maintain smart card specific infrastructure. To use smart card services, host applications must be able to communicate with smart cards. This communication component has been the critical piece of all authentication systems based on smart cards, and is perhaps the reason why smart cards have thus far not enjoyed widespread adoption in security frameworks for ubiquitous and loosely managed systems. In this section, we describe the conventional smart card connectivity model with respect to the X.509 based online authentication, and the usability and deployment issues inherent in the existing methods.

A. *Smart Card Middleware*

Conventional smart cards use traditional ISO 7816 communication protocols to talk to their host devices. These devices range from mobile phone handsets to custom readers at public transportation terminals. In such environments smart cards continue to be useful and well integrated components. Conventional smart cards are also used in online authentication applications, though their acceptance in this market has been less successful. Two key reasons for this are the lack of built-in smart card reader drivers on mainstream PCs, and the need of smart card specific middleware; both of which are barriers for entry into the online market that demands ubiquitous plug-n-play behavior. Although the reader driver issue is addressed in modern computer operating systems (OSs) through the standard USB CCID class driver for smart card connectivity, the distribution of smart card middleware continues to impede the adoption of online authentication solutions.

This middleware enables application programs to access the cryptographic functionalities of smart cards (or other security devices) without worrying about the details of these devices. For this purpose, PC OSs offer a device independent cryptographic API, which is realized by device specific implementations. Different operating systems have their own APIs, and different devices (including smart cards) require their own implementations. Middleware examples include Microsoft's CyptoAPI, RSA Laboratories' PKCS#11, and Apple Computer's CDSA. While offering similar capabilities, they present different APIs and have additional restrictions that may limit the functionalities of applications developed for a specific middleware API. For example, CryptoAPI is used within the native Windows ecosystem [6] but not supported on Mac, Linux, or even browsers other than IE on Windows. The PKCS#11 specification [7], though available across all major operating systems, is natively accessible only via Firefox browser, and is not supported by IE. Similarly, CDSA is only supported on Mac OS X [8].

To further complicate matters, the user may need to manually install these browser/OS specific middleware on all the machines he intends to use. For example, to use Firefox browser with a smart card, the user needs to download and install the PKCS#11 library. Such manual installations severely restrict the portability of smart cards. While the two-factor authentication credentials are stored in a device which you can carry in your wallet, the use of these credentials is not portable. Furthermore, since middleware of a particular smart card may not be available for all browser/OS combinations, it restricts the online authentication solution to a limited number of platforms.

B. X.509 Authentication

The X.509 certificate based authentication utilizes a user’s digital certificate along with the corresponding private key. Using public key cryptography [9] the user can demonstrate that he is indeed the holder of the private key. This can be done by a user storing his private key and using it in calculating response to a server challenge when needed. Unlike passwords which a user can remember, certificates and private keys are blobs of data that need to be securely stored in digital form. To ensure flexibility and inter-operability, the security industry has specified architectures for the storage and use of these credentials either from the operating system of host computer or from an external security device such as a smart card. Since hardware tokens and even software security devices present different interfaces and use different protocols, these architecture specifications provide a common bridge for accessing the cryptographic capabilities of these devices. For example, certificate access, document signature and encryption, and card holder validation can now be done in a device neutral way from a given platform. The middleware mentioned earlier implements some of these specifications. A web browser can use this middleware to accomplish SSL/TLS mutual authentication with the client certificate and private key stored in a smart card. However, the smart card middleware is a local resource; web applications cannot use it to access smart cards in a platform neutral way.

C. Online Authentication Usability

Even if the hurdle of middleware installation is overcome, there can be usability issues. Smart card functionality is accessed via the cryptographic interfaces of web browsers. These interfaces are agnostic to the underlying credential store (smart card, host computer, etc.) and therefore, provide broad abstractions. However, abstractions by their very nature are written at a high level, and seldom address all the specificities of a target device. Because of this, security mechanisms based on smart card conventional connectivity are generally seen as road blocks to application efficiency and often abandoned. Furthermore, certain web browsers, such as IE, require the user to propagate certificates from his smart card to web browser’s persistent certificate store. This makes the smart card based online authentication non-portable by limiting its use to only those computers to which such propagation has been done.

Let us look at another usability aspect by considering the following example:

1. A user browses to a website that requires certificate based authentication.
2. The web browser displays a list of certificates propagated from the user’s smart card.
3. Since each certificate has a specific use, the user is asked to select the appropriate certificate.
4. Once the user selects the certificate he is prompted for a PIN.
5. The user enters the PIN and authenticates successfully.

While this appears simple, Steps 2, 4, and 5 present a user interface challenge. They present the user with a UI that is specific to the browser, host operating system and the smart card middleware. The web application has no control over the way the user interacts with these UI elements. For example, the tasks of canceling the certificate selection, requesting smart card insertion, physically removing the smart card, or abandoning the PIN entry, could provide inconsistent responses. Furthermore, the experience of accessing the same web site varies with each web browser and operating system.

The current smart card connectivity model is therefore not a panacea for achieving a seamless marriage between security and usability. While the notion of carrying your credentials in a secure portable device is a fascinating idea, it fails to germinate into a viable solution that utilizes these credentials for online authentication. We address these issues through a new smart card connectivity method called SConnect, and then show how it can be used to design a smart-card-based online authentication framework.

IV. SCONNECT

SConnect [10] is a connectivity bridge between a smart card and a web application. A web application typically consists of two major components: a server part that executes on a remote web server; and a client part that executes in the local web browser. The server part of the application implements server side business logic, interacts with backend systems, and generates dynamic HTML content to serve the client. The client part of the application renders web content, implements client side logic, interacts with the user, and executes scripts, typically JavaScript. To access the functionality of a smart card connected to a host computer, a web application must communicate with the smart card. SConnect enables this communication, without requiring the installation of any conventional smart card middleware.

A. SConnect Architecture

The SConnect architecture is composed of two parts: a web browser extension; and a library. The web browser extension extends the standard computer and smart card interface layer (called PC/SC) to enable client applications written in JavaScript to communicate with the smart card. The SConnect library provides a JavaScript API for developers to write web applications that connect to and access smart cards. The library uses the browser extension to communicate with smart cards. Figure 1 illustrates the architecture of a SConnect-based web application, with the two shaded boxes representing the two parts of SConnect. Typically the client side JavaScript code in the web application resides on a web server and is downloaded to run in the web browser on demand. Some common code, which interacts with smart cards using the SConnect library, is referred to as smart card module, and is different for each type of smart card. In the conventional approach such differences between smart cards are handled by installing different middleware components, a process that is both difficult to maintain and cumbersome to use. By contrast SConnect allows such support by simply downloading a different JavaScript file, a process that is completely transparent to the user.

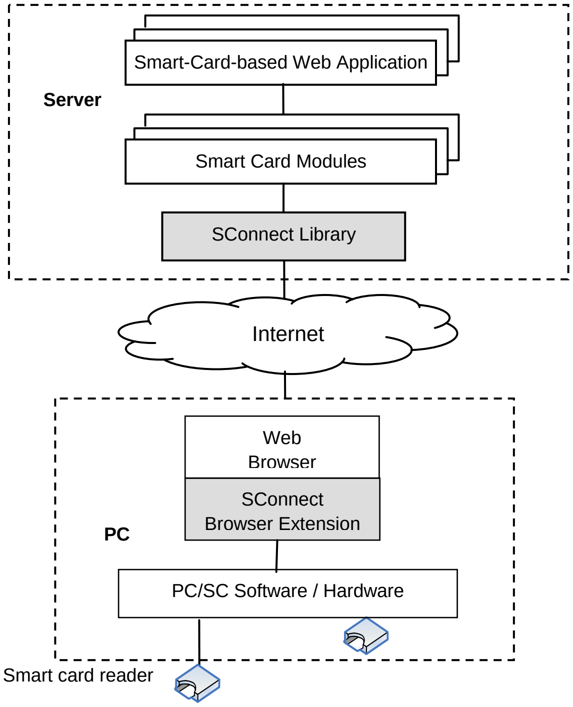


Figure 1. SConnect-based web application architecture.

To ease development, SConnect hides browser dependent complexities from web application developers. The SConnect library provides utility functions that handle the detection, installation, and update of SConnect browser extension. This extension is less than 500KB and is available for most common web browsers on Windows, OS X, and Linux operating systems.

B. SConnect Security Features

While the openness of SConnect that allows direct access to smart cards is a bonanza for web application development, it also broadens the attack surface. Malicious applications can potentially use the same interface to connect to the smart card and use its cryptographic services to impersonate the card holder. To mitigate such potential risks, SConnect deploys a set of security measures to protect the end user and service provider. These measures include digital signature of the browser extension, enforcement of HTTPS, user consent, server verification, and a control mechanism called Connection Key.

Digital Signature: The SConnect browser extension is digitally signed using a code signing key issued by a trusted certificate authority, such as VeriSign. A signed extension instills confidence in users by validating the source of the extension.

Enforcement of HTTPS: To ensure secure communication with a remote web server and to prevent Man-in-the-middle (MITM) attacks, SConnect mandates HTTPS connection between the browser and the remote web server before a web application is allowed to access the smart card. SConnect rejects connection requests from non-HTTPS connections.

User Consent: The first time a user visits a SConnect-enabled website, SConnect displays a warning message box informing the user that the website is trying to access the smart card. The user must make a conscious decision to allow or deny such access. SConnect can save this decision for future reference if so desired by user.

Server Verification: During SSL (or its predecessor TLS) handshaking, the browser receives and examines the server website’s SSL certificate. If this certificate is invalid, the browser presents a warning to the user. However, most users ignore such warnings and continue anyway, thereby exposing themselves to malicious websites and MITM attacks. To mitigate this risk, SConnect does additional server SSL certificate verification when a web application tries to access the smart card. This verification consists of verifying the signatures of the certificate chain, ensuring that the root CA is trusted by the browser, checking the validity period, and matching the Common Name in the certificate with the URL of the website. If SConnect determines that the certificate is invalid, it will not allow any connection between the website and the smart card, even if the user has accepted the browser connection.

Connection Key: While the server verification ensures the identity of a website, it does not make any claims about its trustworthiness. That determination has traditionally been left at the user's discretion - a task that is made even harder by the promiscuous approach to issuance of SSL certificates followed by some certificate authorities, even for the Extended Validation Certificates [11]. To address such risks, and also to introduce a licensing policy, SConnect employs the Connection Key. The authority that issues smart cards can decide which web portals can access these cards and, hence, can issue the Connection Key to only these portals. Examples of such smart card issuing authorities can be governments that issue smart cards to their citizens and want to control at which government service portals that citizens can use these cards.

The Connection Key uniquely binds to the SSL certificate of the website that deploys SConnect-based applications. This ensures that only websites with valid Connection Keys can access the smart card. The Connection Key itself does not contain any secret. It includes a set of attributes such as Common Name (the website domain name), issuer name, issue date, expiration date, and hash of the website's SSL certificate. This information is then signed using the SConnect extension issuer's private key, K_{priv}. The corresponding issuer public key, K_{pub}, is encoded within the SConnect browser extension. SConnect can therefore verify the Connection Key and ensure that the common name in the Connection Key matches the domain name the web browser is currently connected to.

These measures ensure a greater level of trust between the end user and service provider so that the openness of SConnect architecture can be utilized in online applications without reducing the security associated with conventional middleware approaches. The next section describes how this open, yet controlled access is used to design a secure two-factor online authentication framework.

V. TWO-FACTOR AUTHENTICATION

We propose a smart-card-based user authentication method for online access that does not rely on the conventional middleware for connecting to the smart card. Instead it uses SConnect. The authentication is based on a classical challenge-response protocol that uses X.509 certificate and the corresponding private key stored in the user's smart card. What makes this method unique is the benefit it brings to service providers and users alike. Web applications based on this authentication method are easy to develop, deploy, use and maintain.

The authentication software consists of two parts: a server part that resides and runs on the web server; a client part that is dynamically downloaded from the web server, but is executed in the web browser. The server component is responsible for authenticating the user, managing login sessions, logging events, and interacting with certificate authorities or issuers for verifying X.509 certificates. The client component renders the user interface in the web browser for user interaction. It also uses the SConnect extension to connect with the smart card and use its cryptographic services.

When authenticating a user to an online server, located on domain D, the authentication involves the following cryptographic operations:

1. The online authentication server with domain D generates a random challenge $C = \{r, D\}$, which is unique for each authentication request. This challenge is generated by combining a random sequence of bytes r, with the domain of the server, D. The server sends this challenge C to the smart card through the web browser and SConnect.
2. SConnect compares the domain D encoded in the challenge C with the current domain D_b that the browser is connected to. If $D = D_b$, SConnect forwards the challenge to the smart card. Otherwise, SConnect rejects the connection. The authentication fails.
3. If SConnect forwards the challenge C to the smart card, the card digitally signs the challenge using the private key K_{priv}. The resulting signature is the response R:
   $$R = sign\{C\} = RSA_{Encrypt}\{SHA\text{-}1(C)\}K_{priv}$$
4. The response R is sent back to the authentication server along with the X.509 user certificate read from the smart card. The server verifies the signature using the public key, K_{pub} , retrieved from the user's certificate. The computation as follows:
   $$\begin{aligned} H_b &= Decrypt\{R\}\,K_{pub} \\ &= RSA_{Decrypt}\{\,RSA_{Encrypt}\{SHA\text{-}1(C)\}\,K_{priv}\}\,K_{pub} \\ H &= SHA\text{-}1(C) \end{aligned}$$
5. *SHA-1( )* is a cryptographic hash and the chance of collision is therefore extremely small. When $H_b = H$, the authentication server concludes that the user's smart card does indeed hold the private key. The user is authenticated. Otherwise, the authentication fails.

Appending the current domain to the challenge in step 1 helps defend against the Man-In-The-Middle and the chosen protocol attacks. (See Section VI.) The authentication is accurate since it is based on cryptography and there is no uncertainty involved. Figure 2 illustrates the message flow of this authentication process.

The user logs in to a website through the authentication server. This connection is over HTTPS protocol, and the web browser performs server authentication as part of the SSL handshake process. The rest of the numbered steps in user authentication are listed below.

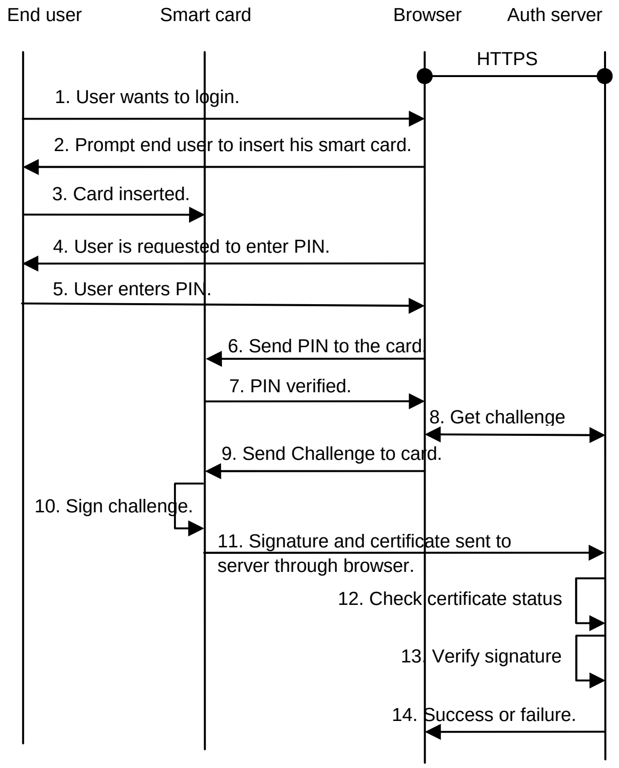


Figure 2. Authentication sequence.

1. The following details the steps in the above sequence:The user clicks the “Login” link in the web page.
2. The authentication client, running in the browser, prompts the user to insert his smart card into a smart card reader attached to the host computer.
3. The user inserts his smart card into the smart card reader.
4. The authentication client prompts the user to enter his PIN in order to use the smart card.
5. The user enters his PIN to the smart card through the web browser user interface or a hardware PIN pad.
6. The authentication client sends the user PIN to the smart card using SConnect communication link.
7. The smart card verifies the PIN, and sends success or failure status back to the browser.
8. If the PIN verification is successful, the client sends a HTTP request to get a challenge from the server. The server responses with a random challenge C that consists of a random number and the server’s domain.
9. The authentication client sends the challenge to SConnect browser plug-in. The latter compares the domain in the challenge with that of the web server to which the browser session is connected. If the two domains are different, SConnect rejects the connection to the smart card. If the two domains are the same, SConnect sends the challenge C to the smart card.
10. The smart card digitally signs the challenge using its private key.
11. The smart card then sends this signature, R, and its X.509 certificate back to the authentication client, which forwards this information to the authentication server.
12. The authentication server verifies the certificate, its issuer, and its revocation status.
13. It also verifies the signature response, R, sent from the card using the public key embedded in the X.509 certificate.
14. If all is good, the server sends a success message to the web browser. Otherwise, the server sends a failure message.

This authentication workflow is simple from the user’s perspective. As evident from Figure 2, the user simply inserts his smart card and enters the PIN. All the details of the X.509 challenge-response handshake for user authentication are hidden from the user. Since there is no classical middleware installation involved, the solution deployment is equally simple for service providers. The challenge for organizations currently using middleware-based smart card authentication solutions is to decide if or not to replace the system with this new approach.

Performance-wise, once the user has inserted the smart card, the time for login is comparable with username/password login, because loading the post-login page typically consumes most of the time.

This proposed method is a two-factor authentication: the what-you-have factor in the form of smart card token (step 3); and the what-you-know factor in the form of user PIN (step 5). It verifies that the user’s smart card indeed holds the private key corresponding to the X.509 certificate. This challenge-response mechanism proves the identity of the user, not whether he has an account at a particular website. The binding of this user identity to a particular account and its access through a given web session are left at the discretion of the service provider web portal.

VI. SECURITY AND USABILITY ANALYSIS

We have presented an authentication framework that is significantly different from the middleware-based authentication architecture used by conventional smart cards. In this section we analyze this framework with respect to protocol security and user behavior.

A. *Protocol Security*

Our authentication method allows a user to login to a remote web server by proving his digital identity to the server using his smart card. From the security perspective, the authentication relies on four complimentary steps to authenticate the user to the server:

1. Server authentication during SSL handshaking, which is done by the web browser.
2. Server verification done by SConnect.
3. Connection key verification done by SConnect.
4. Certificate-based client authentication (as the user authentication).

The first three steps represent a layered approach to authenticate the server. This ensures that the user is interacting with the intended server. Step 1 validates the server's SSL certificate and establishes a secure communication channel with the server. Step 2 is an additional check on Step 1, in case the user ignores the browser warning about an invalid certificate. Step 3 ensures that smart card connectivity is only exposed to websites with valid connection keys. If a website satisfies these three security checks, SConnect allows it to communicate with the smart card. This significantly reduces the attack surface that a typical web application is subjected to. We get the benefits of an open application development model with easy on demand deployment, but without the risk of MITM and other attacks, which we will discuss in more detail below.

Man-In-The-Middle

As the name suggests, Man-in-the-Middle (MITM) acts as a middle person on the network, intercepting messages between a server and a client to gain access to a user's account at the server. For example, the attacker poses as a server S to an unsuspecting client, and then impersonates as the same client to the actual server S. MITM attacks are typically handled through SSL mutual authentication. The smart card stores the client certificate and the corresponding private key. The web browser has a direct access to the client certificate and the operations that use the private key. The smart card specific middleware discussed earlier in Section III-A makes such access possible. While this approach certainly provides a more robust security model, it is at the expense of usability.

In our proposed authentication method, client authentication is done at the application level, after the SSL handshake and SConnect have verified the server. Although this two-step approach to mutual authentication by itself is vulnerable to MITM, the potential risks are mitigated by the security checks performed by SConnect. Assume an attacker, a malicious website www.bad.com, is acting as MITM between a client and a legitimate server www.good.com through, for example, DNS poisoning or some other means. This MITM attack is addressed as follows:

1. SConnect server verification will fail since the browser has connected to www.good.com, while the common name in SSL server certificate is www.bad.com. SConnect will catch this mismatch even if the user has ignored the browser warning.
2. SConnect uses the web browser's root certificate store to verify the validity of an SSL certificate. In case the attacker uses a self-signed certificate whose issuer certificate is not in the root certificate store, or uses a fake certificate for www.good.com, SConnect server verification will catch the error because it cannot verify the certificate. It will reject the connection even if the user has ignored the browser warning.
3. In the unlikely event that the attacker obtains a valid SSL certificate issued by a trusted CA in the name of www.good.com and the corresponding private key, SConnect will refuse access to smart card unless the attacker also presents a valid Connection Key. The attacker may copy the Connection Key issued to the actual www.good.com, but it still cannot pass the Connection Key verification because the thumbprint of its SSL certificate is different from that of the fake www.good.com SSL certificate.

This layered approach to security allows our authentication framework to offer mutual authentication using a two-stage protocol. While it may not be as secure as a monolithic mutual authentication protocol such as SSL, it offers an excellent balance between security and convenience.

Chosen Protocol Attack

In a chosen protocol attack, the attacker lures the user into using his authentication credential at a malicious website when the same credential can be used on a legitimate website [12]. For example, a user has an account at an online bank, www.bank.com, which supports the smart-card-based authentication. An attacker could lure the user into authenticating to a malicious website using the same smart card. During the authentication process, the attacker can login to the user's account at the bank by forwarding the challenge from the bank to the smart card and the response from the smart card to the bank. In order to do so, the attacker must have a valid SSL certificate and a valid Connection Key. This could happen if an otherwise legitimate website either turns malicious or is temporarily compromised.

The domain information added to the server challenge will prevent such an attack. For example, the domain name www.bank.com is a part of the server challenge for user authentication. The attacker website forwards the challenge to the smart card. There are two possibilities. First, if the attacker changes the domain name to its own, SConnect verification will pass and the smart card will generate a response. However, the response verification on www.bank.com server will fail and so will the authentication. This is because the domain name in the response is different from the actual domain name of www.bank.com. Second, if the attacker does not change the domain name, the SConnect verification will fail because the domain name in the challenge is different from the current domain. SConnect will reject the connection to the smart card.

B. User Behavior

Modern web browsers check the SSL server certificate when establishing an HTTPS connection with a given website. The purpose of this check is to ensure that the SSL certificate is issued by a trusted certificate authority, the certificate's Common Name (CN) matches the website's URL, and that the certificate has not expired. If any of these assertions fails, the browser informs the user that the certificate is not valid and recommends that user not connect to the given website. However, this is only a recommendation, and browsers will still proceed with the connection if the user ignores this warning. Surprisingly, such warnings are not rare. A survey of Internet use published in 2007 found that roughly two-third of all SSL certificates used for secure connections generated warnings [13]. No wonder users have become accustomed to seeing these SSL warnings and casually ignore them. Our authentication framework uses SConnect server verification to strictly enforce what browsers merely recommend.

VII. CONCLUSIONS

Achieving usable security is very challenging. While smart cards offer unparalleled hardware security, their applicability has been restricted to tightly controlled environments where smart card infrastructure can be managed. This paper introduced a new online authentication framework that is different from other smart-card-based authentication solutions. It works within the existing smart card infrastructure, but still offers a truly plug-n-play experience users have come to expect from web applications. Instead of relying on pre-installed middleware, the new framework uses a browser based approach to access smart card services. This not only provides a more familiar user interface, but also allows online service providers to deploy and update their service offerings without requiring the user to install a new application. Our future work for enhancing this authentication framework will focus on practical issues such as; Connection Key revocation, management of SConnect browser extension updates, and handling of SConnect browser extensions that are issued by different authorities.

With this technology, we foresee a new trend in the development of smart card based Internet security solutions that go well beyond user authentication. Additional security services such as email encryption, document signature, secure transactions, etc. can be delivered on demand using the familiar web browser interface.

REFERENCES

[1] "Gmail, Yahoo Mail join Hotmail; passwords exposed", ComputerWorld, http://www.computerworld.com/s/article/9138956/Microsoft_confirms_phishers_stole_several_thousand_Hotmail_passwords. (last access 07/13/2011).

[2] Byron Acohido, Hackers breach Heartland Payment credit card system. USA Today, http://www.usatoday.com/money/perfi/credit/2009-01-20-heartland-credit-card-security-breach_N.htm, January 2009. (last access 07/13/2011).

[3] Carrie-Ann Skinner, One-Third Use a Single Password for Everything, PCWorld, http://www.pcworld.com/businesscenter/article/161078/onethird_use_a_single_password_for_everything.html, March 2011. (last access 07/13/2011).

[4] Frederik Mennes, Best Practices for Strong Authentication in Internet Banking, ISSA Journal, December 2007. http://www.issa.org/Library/Journals/2007/December/Mennes-Best%20Practices%20for%20Strong%20Authentication%20in%20Internet%20Banking.pdf. (last access 07/13/2011).

[5] Smart Card Alliance, "What makes a smart card secure?," A Smart Card Alliance Contactless and Mobile Payments Council White Paper, CPMC-08002, October 2008. http://www.smartcardalliance.org/resources/lib/Smart_Card_Security_WP_20081013.pdf. (last access 07/13/2011).

[6] Microsoft, Cryptographic Service Providers, http://msdn.microsoft.com/en-us/library/ms953432.aspx. (last access 07/13/2011).

[7] RSA Laboratories, PKCS#11: Cryptographic Token Interface Standard, http://www.rsa.com/rsalabs/node.asp?id=2133. (last access 07/13/2011).

[8] Apple, Mac OS X Security Framework.

[9] C. Adams and S. Lloyd, Understanding PKI: concepts, standards, and deployment considerations, Addison-Wesley Professional; 2nd edition, Nov. 2002.

[10] Kapil Sachdeva, H. Karen Lu, and Ksheerabdhi Krishna, "A browser-based approach to smart card connectivity," IEEE Workshop on Web 2.0 Security and Privacy, Oakland, California, May 21, 2009.

[11] CA/Browser Forum, Guidelines for the Issuance and Management of Extended Validation Certificates, Version 1.0, 7 June 2007, http://www.cabforum.org/EV_Certificate_Guidelines.pdf. (last access 07/13/2011).

[12] R. Anderson, Security Engineering, 2nd edition, Wiley Publishing, Inc., 2008.

[13] C. Jackson and A. Barth, ForceHTTPS: protecting high-security web sites from network attacks, Proceedings of WWW 2008, Apr. 2008, Beijing, China.

Complex Event Processing for Usage Control in Service Oriented Infrastructures

Alexander Wahl, Stefan Pfister, Bernhard Hollunder
Department of Computer Science
Hochschule Furtwangen University
Furtwangen, Germany
alexander.wahl@hs-furtwangen.de, stefan.pfister@hs-furtwangen.de, bernhard.hollunder@hs-furtwangen.de

***Abstract*—Service oriented architectures (SOA) should adhere to clearly defined quality attributes, which are formalized using policies. Well-known attributes in the security realm are access control and usage control. Our approach is to analyze operations (e.g., data deletion) and data flows that occur within a SOA. We use the extracted information to monitor policies, especially usage control policies. We focus on usage control, which is by far not as well investigated as access control, but highly relevant for providers of sensitive data who do not want to lose control on their data. We show that there is a transformation that maps usage control formulas, formalized in an appropriate policy language, to rules based on Complex Event Processing (CEP) technology. We further argue that by the combination of a policy language, the CEP technology, sensor components and a transformation from policy language to CEP rules SOA infrastructures can be enabled for usage control.**

Keywords - Service Oriented Architecture; Web service; Usage Control; Complex Event Processing;Policies.

I. INTRODUCTION

Today, data are commonly exchanged in distributed computer systems. For some of these data stakeholders have increasing interest to control access to these data and to further control the usage of data once they are distributed. For example, medical data of patients generated by physicians during treatment are to be highly protected. Such data may be accessed by authorized persons only, like other physicians, but may usually not be handed over to any others without permission of the patient. In Germany this is regulated by law, e.g., the so-called "Bundesdatenschutzgesetz" [1]. Access to data is covered by access control, but to prohibit propagation of data a concept for usage control is required.

Access control deals with the question: Who may access data at first instance? For access control, there are well-known approaches, such as the role based access control (RBAC) principle [2].

A. Usage control

Usage control [3] deals with the question: what happens to data once they are given away? Distributed usage control [4] is an extension of usage control in distributed systems.

There are two main parties in usage control: *data providers* and *data consumers*. A data provider owns data and controls access to them. The data consumer wants to gain access to and perform operations on these data. Once the access is granted, data are handed over to the data consumer. Without usage control the data provider from then on has lost control on his data. The data consumer may perform unwanted operations on the data of which the data provider may not get informed. For example, confidential information intended to be used within a company and its suppliers would be passed to a competitor without being noticed. With usage control the data provider regains control on his data. He may now specify usage control rules.

Usage control rules describe the conditions a certain operation on data is allowed or prohibited. Such conditions are either provisions or obligations. Provisions are those that refer to the past and the present, respectively whether data may be released in the first place. Conditions that govern the present and future usage of the data are so-called obligations [5]. Typical examples for usage control rules are "Delete a particular document within 30 days", "Do not give data D to anybody else", "Data D may be copied at most 2 times", "Data D may only be sent if contract exists" or "Using data D for some purpose requires an acknowledgement of data provider". Usage control rules are of different types, as can be seen with the given examples. They either relate to time, cardinality, environment, purpose or occurrence of events [6].

A *usage control policy* is a formal representation of a usage control rule. It consists of usage control formulas. To describe usage control policies in a formally correct manner usage control policy languages were introduced, like e.g., Obligation Specification Language (OSL), Usage Control (UCON) and Extended Privacy Definition Tool (ExPDT). We will get back to these later.

ISBN: 978-1-61208-152-6

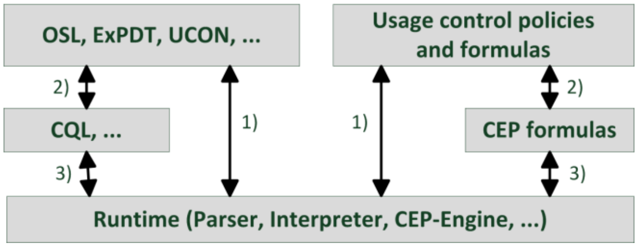

OSL, ExPDT, UCON, ...
Usage control policies and formulas
2)
1)
1)
2)
CQL, ...
CEP formulas
3)
3)
Runtime (Parser, Interpreter, CEP-Engine, ...)

Figure 1. Mapping of policies to runtime. 1) Direct mapping; 2) Intermediate mapping 3) Mapping to runtime.

Once usage control policies are formulated and applied to data, the infrastructure is in charge to ensure compliance to that sets of formulas, i.e. policies. That means that once a usage control formula is violated the execution of an operation should inform the data provider. Some usage control formulas, like “Do not give data D to anybody else” or “Data D may be copied at most 2 times”, may even hinder the execution of an operation (e.g., copying the data) and return a fault.

B. Using Complex Event Processing for Usage control

For usage control, especially for distributed usage control, the availability of appropriate technologies is still very limited. There are a few approaches on usage control, as we will see in section “Related Work”. They are either prototypic, proprietary, or they are limited to SOA infrastructures that contain specific components, such as enterprise service bus (ESB). All these approaches have in common that they enforce policy formulas, defined in a specific policy language, directly to the runtime system (see Figure 1). We think, that by introducing a solution based on a well-established and well-tested technology, like Complex Event Processing (CEP) [7], we can overcome the issues described before. We further argue that the usage of CEP simplifies the mapping from policy language formulas to a runtime environment.

In this work, we combine a usage control policy language, the CEP technology, sensor components and a transformation of usage control formulas to CEP rules. The novelty of our approach is that instead of mapping a policy directly to the runtime, e.g. by the usage of a proprietary interpreter [8], we introduce an intermediate step that maps a policy to CEP rules. The main advantages are the usage of a well-established and well-tested technology and the less complexity of mapping to the runtime, since the evaluation of CEP rules is performed by an already existing component, namely the CEP engine.

We use the Continuous Query Language (CQL) [9] to formulate the CEP rules. With CQL, the formula is still comprehensible once the formulas are transformed. The advantage is that the mapping from CEP rules to a runtime environment already exists. In addition our solution can be used to enable existing SOA infrastructures for usage control with minor modifications only. Our approach is based on the overall architecture described in [10].

This paper is structured as follows: Section 2 gives an overview on related work in the area of usage control. In Sections 3 we describe our approach in detail. In Section 4 we describe a strategy to transform usage control formulas into a technical representation. We further discuss events and show how CEP rules are evaluated based on events. We will also show that there is a direct relationship between formulas, events and necessary information to be extracted from the SOA infrastructure. Section 6 is about the relationship between CEP rules and policies. In the last section, we will conclude our work and give a forecast to our future work.

II. RELATED WORK

In this section, we will describe the most significant publications on usage control, usage control models and usage control policy languages.

Hilty, Pretschner and coauthors gave, in a series of publications, a detailed overview on enforcement of usage control [11, 12] and distributed usage control [4]. They introduce a usage control policy language called Obligation Specification Language (OSL) [6], monitors for OSL-based usage control [8] and usage control in service oriented architectures (SOA) [13]. OSL enables to formulate temporal descriptions of obligations. In “Monitors for Usage Control” a prototypical implementation of a Java based obligation monitor for OSL is mentioned [8]. In “Usage Control in Service-Oriented Architectures” they stated, that “Implementing the architecture is a next step” [13].

Park and Sandhu introduced the concept of usage control [14] and the ABC-Model for usage control ($UCON_{ABC}$) [3, 15]. This model integrates *A*uthorization (A), o*B*ligation (B) and *C*ondition (C) components. The latter, conditions, are environmental restrictions before or during usage of data. However, to our best knowledge they did not implement their approach for a SOA infrastructure.

Gheorghe et al. implemented a policy enforcement mechanism on ESB level [16]. They called it xESB, which is an ESB enhanced by an additional component based on Java Business Integration (JBI) [17]. First they used a specific policy language, but in a following publication they enforce UCON policies, respectively POLPA [18], which is used to implement the UCON model. This elegant approach is limited to SOA infrastructures using ESB technology. However, there are SOA infrastructures that are not implemented from scratch and that do not use ESB. And for those this approach is not applicable.

Kaehmer et al. [19–22] introduce ExPDT. With ExPDT, permissions, prohibitions and orders based on contextual conditions or obligations can be described. It enables for access and usage control, and also supports the comparison of two policies.

To sum up, there are several expressive policy languages for usage control. With these policy languages obligations, conditions, permissions, prohibitions and orders can be formalized, depending on their expressiveness. Also a few implementations to monitor usage or to enforce usage control in infrastructure are available. However, these are either prototypic, limit themselves by requirements on the existing SOA infrastructures, and use more complex transformations from the policy language to the runtime.

III. Usage Control Policy languages

To build a SOA and to apply usage control to it is a quite sophisticated task. If a SOA is built from scratch usage control mechanisms can be taken into account from the very beginning. An appropriate approach, like xESB, can be chosen. The same approach can be used for SOA infrastructures that already use an ESB. But in real world, most often there is an existing SOA that is to be enabled to usage control, as described in the example of the Hong Kong Red Cross by [23]. But unfortunately not all SOA infrastructures do use an ESB, and for those alternative approaches are needed.

Within an existing SOA based on Web service technology, service providers and service consumers exchange data. Thereby some of these data might be sensitive ones, like e.g., confidential patient data within a hospital, or data for internal use only within a company. To achieve usage control, such data are associated with usage control policies. Well-known candidates for specifying declarative policies are (among others): eXtensible Access Control Markup Language (XACML) [24], Extended Privacy Definition Tool (ExPDT) [20] (compliance validation, privacy preferences, permissions, prohibitions), Usage Control (UCON) [14], and Obligation Specification Language (OSL) [6].

XACML offers a policy language to describe general access control requirements. It offers a request/response language to determine whether an action is allowed or not. The focus of XACML is on access control. However, to a certain extend it can be used for usage control, especially if enhanced by additional features. U-XACML [25], for example, enhances XACML with UCON features.

OSL supports the formalization of a wide range of usage control requirements. It mainly focuses on obligation. The language contains propositional operators (AND, OR, NOT, IMPLIES), temporal operators (UNTIL, AFTER, DURING, WITHIN), cardinality operators (REPUTIL, REPMAX) and permit operators (MUST, MAY).

UCON is a general model for usage control. With UCON_{ABC} a policy specification is provided to support pre- and post-authorization, obligation rules and conditions.

Finally, ExPDT is a policy language developed to define privacy preferences. It allows describing permissions, prohibitions and orders that are to be followed once certain contextual conditions are met or if obligations have to be fulfilled.

In this work, we will limit to a transformation from OSL formulas to CEP rules (due to the restricted number of pages). The expressivity of OSL is sufficient for the examples used in this work. Transformations from other languages, like ExPDT, will be future work.

IV. Approach

In general, a SOA consists of multiple collaborating entities: i) service providers and service consumers, ii) data providers and data consumers, iii) infrastructure, iv) data and v) events. Service providers offer some functionality that is utilized by service consumers. A service consumer itself may be a service provider for another service consumer (transitivity). Similar to that, data providers offer data to data consumer. Transitivity also applies here. The infrastructure is the collectivity of system components, frameworks, applications, etc. necessary to run the SOA.

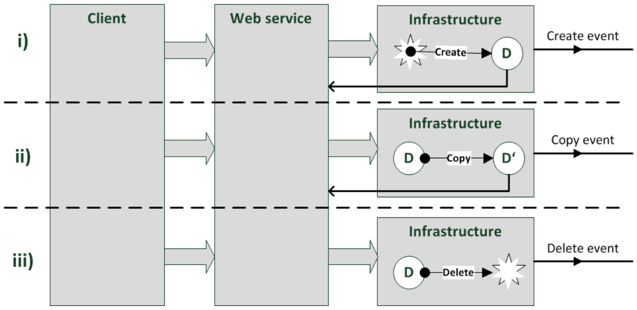


Figure 2. Operations on data produce events

Data are information sets that are generated, copied and deleted at the infrastructure (see Figure 2). Creating data means that information is passed to and stored within the infrastructure, e.g., adding a dataset for a person to a database. Copying data means, that data existing within the infrastructure is duplicated and sent elsewhere. For example, copied data are exchanged between data provider and consumer within the body of a message. Deleting data describes removing of data from the infrastructure. In any case the infrastructure is involved at any operation that is executed on data, and is therefore potentially able to inform about these operation.

An alternative to inform a third party, e.g., a usage control monitor, about an operation on data is the following one: At an existing SOA infrastructure so-called sensor components are applied at appropriate entities, e.g., a SOAP message handler attached to a Web service, a JBI component for an ESB (if part of the SOA), or a sniffer that analyses the traffic at the application servers port (see also [10]). The aim of the sensor component is i) to detect operations executed on data (either before or after the execution), and ii) to inform a third party by emitting an event.

In the context of sensor components an *event* is a message that identifies the executed operation and the affected data. It contains a timestamp and additional data, e.g., the principal of an operation. The event is emitted by the sensor component and received by the third party, like the usage control monitor. Within the usage control monitor the included formulas are evaluated based on the received events.

The usage control formulas to be evaluated are specified by a developer via a usage control policy language. Afterwards they are transformed by the developer to a representation the monitor is able to interpret.

For example: A service consumer calls a Web service in order to get a copy of a certain data. The Web service therefore calls the infrastructure for a copy of this data. Within the infrastructure, each time a data is copied a sensor component is involved prior to copying. This sensor component produces a copy event. This event, for example, includes information that identifies the data to be copied, a timestamp and principal of the copy request. The copy event is then emitted by the sensor component and received by a usage control monitor.

The usage control monitor is responsible to evaluate if the usage control formulas are satisfied. The usage control monitor, in essence, is the CEP engine equipped with formulas as required. The evaluation is performed based on the events received by the CEP engine.

V. STRATEGY

In the previous section, we described an approach that uses events (emitted by sensor components that are attached to the SOA entities) to evaluate a CEP rule, i.e., a transformation of a usage control policy. In this section we will consider in more detail the transformation from a usage control formula, specified in a policy language, to a CEP rule (Figure 1). We will further illustrate how a CEP rule is evaluated based on events. We will also show that there is a correspondence between CEP rules and the events that are necessary to determine if a CEP rule is satisfied, i.e., a corresponding usage control policy is satisfied.

The notion "satisfied" in the context of usage control policy means that its related formulas are fulfilled. In the context of CEP rules satisfied means that based on the collected events the preconditions of a CEP rule evaluate to true and the CEP rule fires. If a CEP rule fires a corresponding event is generated and emitted to a subsequent actor (see Figure 3). The actor then initiates a corresponding action, e.g., a deleting data.

Usage control is applied to a SOA by binding usage control policies to data using the sticky policies paradigm [26]. Each formula follows a pattern similar to "if <condition> then <action>" or "if <condition> then (not) <usage>" [6]. For example, assume the usage control formula "Delete document within 30 days". This formula can be reformulated: "if document D will not be deleted within 30 days then indicate violation". In OSL this formula is specified as follows:

```
Within(30days,delete,{(data,D)})
```

In this example *delete* specifies the event, *data* specified a parameter and *D* the value of the parameter *data*. The transformation of this formula to a CEP rule is based on the following considerations:

1) This formula specifies time duration of 30 days. So we have two points in time: a creation timestamp and deletion timestamp.
2) We want to get informed if the formula is violated.

The first consideration implies that we need to get informed on the creation of data and on the deletion of data. From that information we can derive that we will need two events to evaluate satisfaction of this formula: *create(D)* and *delete(D)*. We suppose that D is not copied in the meantime. *Create(D)* is emitted on data creation, *delete(D)* on data deletion.

As we already described, the infrastructure does have the potential to inform about an operation that is performed on data. So it needs to be enabled to emit events. We therefore modify the infrastructure by adding sensor components at appropriate positions.

A sensor component in brief is a piece of software that is attached to the SOA infrastructure at appropriate SOA entities. Sensor components can be of different types. They can be message handlers (e.g., SOAP message handlers), JMX client components, sniffers or even GUI elements. These sensor components have in common, that they collect and analyse actual data within the SOA infrastructure and emit these data as events. For a more detailed description please refer to [10].

Since sensor components are additional components one has to expect certain performance penalties once they are applied. However, the performance penalty for extracting data and emitting an event should be small. And since the expensive (in terms of execution time) evaluation of CEP rules can be performed in asynchronous manner by a third party, e.g. a dedicated machine running a CEP engine, the influence on performance can be kept to a minimum.

In our example we need to apply a sensor component that emits *create* events, and a second one that emits *delete* events. The events emitted by the sensor components are defined by a developer. His task is to analyze i) which information is necessary to evaluate the CEP rule, ii) from where in the SOA the information can be fetched. Based on this he defines the event types. Figure 2 shows a few exemplary event types. There are several other events one can think of, also complex events. For example the event *consumption(D)*, which might be defined as *create(D) OR copy(D) → consumption(D)*. In words, each time D is created or copied, a consumption event is produced. With that further usage control formulas can be formulated, like "IF *consumption(D) THEN check(contract)*" or "IF *consumption(D) THEN inform(data provider)*".

The events are collected by corresponding event streams in the CEP engine, i.e. the input of a CEP rule. Each CEP rule has its own set of event streams. By these streams the CEP rule defines which events must be emitted by the infrastructure and the sensor components, respectively. So, for each CEP rule there is a well-defined set of events to evaluate its satisfaction. In consequence, there is a direct relation between CEP rules and events. A CEP rule is transformed from a usage control policy formula. Therefore the relation to events also applies the latter (and finally to the verbal formulation of them).

In the example "Delete document within 30 days" stressed before there are two event streams, namely: CreateStream (contains all collected events *create(D)*) and DeleteStream(contains all events *delete(D)*). These two event streams are the input of our CEP rule. An instance of an event *create(D)* is named createEvent, respectively deleteEvent for *delete(D)*.

As considered previously, we want to get informed on violation. By the way, it is also possible to indicate that the formula is satisfied. In either case we can again use events to denote violation or satisfaction. In our example we define an event *violation(D)* that is emitted by the CEP rule once the formula is not satisfied. In other words, the output of our CEP rule is an instance of *violation(D)*, namely violationEvent.

A CEP rule defines how events are correlated. Consider the OSL formula "Within(30days,delete,{(object,D)})".

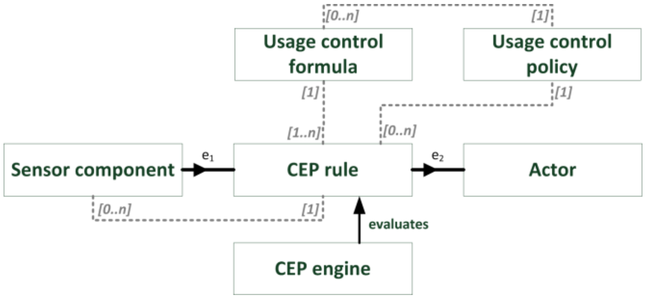


Figure 3. Relations between CEP rules and usage control policies.

Based on the information of the former text this policy can be translated into a CEP rule:

```
SELECT createEvent, deleteEvent
FROM CreateStream, DeleteStream
WHERE createEvent.Data = deleteEvent.Data
AND deleteEvent.Time - createEvent.Time > 30 days
THEN CREATE violationEvent
```

In other words: createEvents from CreateStream and deleteEvents from DeleteStream that operate on same data, and whose timespan between creation and deletion is greater than 30 days cause an violationEvent to be created and emitted.

It is obvious, that if events of *create(D)* and of delete(D) are not emitted by sensor components the formula cannot be evaluated and fire a violationEvent.

Summing up, a SOA infrastructure has to be enabled for CEP based usage control. Therefore, sensor components have to be installed in the SOA at appropriate positions. Sensor components analyze information within the SOA and emit events. Events are collected and correlated to evaluate CEP rules. There is a direct relation between a CEP rule and the events needed. Since the CEP rule is a transformation of a policy language formula this relation also applies to the latter.

VI. CEP RULES AND POLICIES

Between usage control policies, usage control formulas and CEP rules there are several relations, as depicted in Figure 3.

First, a usage control policy is a set of usage control formulas, i.e., it consists of zero or more formulas. A usage control policy is (usually) satisfied iff all of its usage control formulas are satisfied.

A usage control formula can be described by one or more CEP rules. So a usage control formula is satisfied iff all the corresponding CEP rules are satisfied.

If a CEP rule is satisfied a corresponding events is generated and emitted. The event (e_2 in Figure 3), or a set of events, is used to trigger a related actor that executes a related action.

A CEP rule is evaluated by the CEP engine based on a set of events (e_1 in Figure 3). So for each CEP rule the number and kinds of events that are necessary to evaluate it is known.

Since the kinds of events to evaluate a CEP rule are known, the sensor components that need to be installed in the SOA infrastructure is also known.

Finally, a usage control policy can be represented by a set of CEP rules. That is because a usage control policy consists of a set of usage rules, and a set of usage control rules can be represented by a corresponding set of CEP rules. So, a usage control policy is satisfied iff a corresponding set of CEP rules.

VII. CONCLUSION AND FUTURE WORK

A. *Summary*

The mapping of usage control formula to the runtime is a difficult and complex task. The introduction of an intermediate step, as shown in Figure 1, is reasonable and brings advantages. By using CEP, the mapping of a usage control policy to the runtime is reduced to mapping to a CEP rule. In this work we mapped an exemplary OSL formula to CEP rule. However, we think that this is also feasible for other policy languages, which is future work. This mapping can be performed more easily. The satisfaction of a CEP rule to runtime is determined by the CEP engine. However, it is necessary to install sensor components in the SOA infrastructure. With these sensor components the SOA is enabled to emit events on operations.

Using CEP the requirements to an SOA infrastructure and the necessary changes within to enable for CEP are kept to a minimum. The approach does not require special CEP components, like e.g., ESB, to be applicable. It is flexible to apply to a variety of SOA infrastructures. Just sensor components need to be applied. The sensor components and events needed to evaluate satisfaction of a CEP rule are in a direct relation. However, currently the sensor components need to be implemented, configured and installed manually.

With CEP not only single formulas can be mapped, but also whole policies. This is interconnecting CEP rules with each other.

B. *Perspective*

The perspective of our work is to enrich existing systems by Quality of Service (QoS) attributes insufficiently supported or yet unsupported at all. We see usage control as one of these QoS attributes. Based on an architecture described elsewhere [10] we currently implement exemplary usage control formulas using CEP technology, and the appropriate sensor components.

We further work on an implementation of a tool chain that supports developers to equip existing SOA infrastructures with QoS attributes [27]. Also a part of these efforts is to automate the transformation from a policy (formula) to a CEP rule. Beside these steps we also plan to include further analysis, like the influence of sensor components on the performance.

Finally, the statements within Section 6 need to be formulated and proved in a more formal manner. Also, this will be part of our future work.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for giving us helpful comments. This work has been partly supported by the German Ministry of Education and Research (BMBF) under research contract 17N0709.

REFERENCES

[1] Bundesministerium der Justiz, Bundesdatenschutzgesetz (BDSG). Available: http://www.gesetze-im-internet.de/bdsg_1990/index.html, last accessed 2011, June 27.

[2] M. Benantar, *Access Control Systems*: Gardners Books, 2010.

[3] J. Park and R. Sandhu, "The UCONABC usage control model," *ACM Trans. Inf. Syst. Secur*, vol. 7, pp. 128-174, 2004.

[4] A. Pretschner, M. Hilty, and D. Basin, "Distributed usage control," *Commun. ACM*, vol. 49, pp. 39-44, 2006.

[5] M. Hilty, D. Basin, and A. Pretschner, "On Obligations," in *Lecture Notes in Computer Science, Computer Security – ESORICS 2005*, S. Di Vimercati, P. Syverson, and D. Gollmann, Eds.: Springer Berlin / Heidelberg, 2005, pp. 98–117.

[6] M. Hilty, A. Pretschner, D. Basin, C. Schaefer, and T. Walter, "A Policy Language for Distributed Usage Control," in *Lecture Notes in Computer Science, Computer Security – ESORICS 2007*, J. Biskup and J. López, Eds.: Springer Berlin / Heidelberg, 2007, pp. 531–546.

[7] D. C. Luckham, The power of events: An introduction to complex event processing in distributed enterprise systems. Boston: Addison-Wesley, 2002.

[8] M. Hilty, A. Pretschner, D. Basin, C. Schaefer, and T. Walter, "Monitors for Usage Control," in *IFIP International Federation for Information Processing, Trust Management*, S. Etalle and S. Marsh, Eds.: Springer Boston, 2007, pp. 411–414.

[9] A. Arasu, S. Babu, and J. Widom, "The CQL continuous query language: semantic foundations and query execution," *The VLDB Journal*, vol. 15, pp. 121-142, http://dx.doi.org/10.1007/s00778-004-0147-z, 2006.

[10] A. Wahl, A. Al-Moayed, and B. Hollunder, *An Architecture to Measure QoS Compliance in SOA Infrastructures.* Available: http://www.thinkmind.org/index.php?view=article&articleid=service_computation_2010_2_10_20064, last accessed 2011, June 27..

[11] Manuel Hilty and Er Pretschner et al, Enforcement for Usage Control- An Overview of Control Mechanisms Deliverables 1 and 2 DoCoMo Euro-Labs Publication, last accessed 2011, June 27.

[12] A. Pretschner, M. Hilty, F. Schutz, C. Schaefer, and T. Walter, "Usage Control Enforcement: Present and Future," *Security Privacy, IEEE, title=Usage Control Enforcement: Present and Future*, vol. 6, no. 4, pp. 44–53, 2008.

[13] A. Pretschner, F. Massacci, and M. Hilty, "Usage Control in Service-Oriented Architectures," in *Lecture Notes in Computer Science, Trust, Privacy and Security in Digital Business*, C. Lambrinoudakis, G. Pernul, and A. Tjoa, Eds.: Springer Berlin / Heidelberg, 2007, pp. 83–93.

[14] J. Park and R. Sandhu, Eds, Towards usage control models: beyond traditional access control.

[15] R. Sandhu and J. Park, "Usage Control: A Vision for Next Generation Access Control," in *Lecture Notes in Computer Science, Computer Network Security*, V. Gorodetsky, L. Popyack, and V. Skormin, Eds.: Springer Berlin / Heidelberg, 2003, pp. 17–31.

[16] G. Gheorghe, S. Neuhaus, and B. Crispo, "xESB: An Enterprise Service Bus for Access and Usage Control Policy Enforcement," in *IFIP Advances in Information and Communication Technology, Trust Management IV*, M. Nishigaki, A. Jøsang, Y. Murayama, and S. Marsh, Eds.: Springer Boston, 2010, pp. 63–78.

[17] Sun Java Community Process Program, *Sun JSR-000208 Java Business Integration.* Available: http://jcp.org/aboutJava/communityprocess/final/jsr208/index.html, last accessed 2011, June 27.

[18] F. Baiardi, F. Martinelli, P. Mori, and A. Vaccarelli, "Improving Grid Services Security with Fine Grain Policies," in *Lecture Notes in Computer Science, On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops*, R. Meersman, Z. Tari, and A. Corsaro, Eds.: Springer Berlin / Heidelberg, 2004, pp. 123–134.

[19] Martin Kähmer and Maike Gilliot, *Extended Privacy Definition Tool.* Available: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-328/paper12.pdf, last accessed 2011, June 27.

[20] M. Kahmer, M. Gilliot, and G. Muller, Eds, *Automating Privacy Compliance with ExPDT*, 2008.

[21] M. Kahmer and G. Muller, Automating Privacy Compliance for Personalized Services.

[22] M. Kähmer, ExPDT: Vergleichbarkeit von Richtlinien für Selbstregulierung und Selbstdatenschutz, 1st ed. Wiesbaden: Vieweg + Teubner, 2010.

[23] G. Gheorghe, B. Crispo, D. Schleicher, T. Anstett, F. Leymann, R. Mietzner, and G. Monakova, "Combining Enforcement Strategies in Service Oriented Architectures," in *Lecture Notes in Computer Science, Service-Oriented Computing*, P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds.: Springer Berlin / Heidelberg, 2010, pp. 288–302.

[24] eXtensible Access Control Markup Language (XACML), 2005.

[25] M. Colombo, A. Lazouski, F. Martinelli, and P. Mori, "A Proposal on Enhancing XACML with Continuous Usage Control Features," in *Grids, P2P and Services Computing*, F. Desprez, V. Getov, T. Priol, and R. Yahyapour, Eds.: Springer US, 2010, pp. 133–146.

[26] G. Karjoth, M. Schunter, and M. Waidner, "Platform for enterprise privacy practices: privacy-enabled management of customer data," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, Berlin, Heidelberg: Springer-Verlag, 2003, pp. 69-84.

[27] B. Hollunder, A. Al-Moayed, and A. Wahl, "A Tool Chain for Constructing QoS-aware Web Services," in *Performance and dependability in service computing: Concepts, techniques and,* V. C. E. Cardellini, K. R. L. J. C. Branco, J. C. Estrella, and F. J. Monaco, Eds, Hershey: Information Sci Refer Igi, 2011.

Semantic Web Service Process Mediation in WSMO:

Current Solutions and Open Issues

Kanmani Munusamy, Mohd Sapiyan Baba
Faculty of Computer Science & Information
Technology,
University Malaya (UM),
Kuala Lumpur, Malaysia
{kanmani, pian}@um.edu.my

Suhaimi Ibrahim, Harihodin Selamat, Keyvan
Mohebbi, Mojtaba Khezrian
Advanced Informatics School (AIS),
Universiti Teknologi Malaysia (UTM),
Kuala Lumpur, Malaysia
{suhaimiibrahim@, harihodin@, mkeyvan2@live,
kmojtaba3@live}.utm.my

***Abstract*—Process mediation plays an important role in ensuring successful interaction between a provider and a service requestor. Therefore process mediation could be conceptualized as a 'middleware' that coordinates the interaction between web services. The Semantic Web Service promises automation in discovery, selection and composition but is still facing serious challenges in resolving mismatches where the Web service interaction takes place. For this paper, the WSMO, a Semantic Web Services framework is chosen and the current process mediation approaches that have adopted this framework are analyzed. The findings enable the identification of some open issues and process mediation elements. These identified factors can be further explored to support automatic communication mismatches in the generic Web Services.**

Keywords-Semantic Web Service; Process Mediation; Communication Mismatches; Mismatch Patterns; Choreography Interface

I. INTRODUCTION

Web service is one of the rapidly growing technologies that have been widely adopted by many organizations in industry. The main goal of the web service is to produce software component and business application that are available via the standardized interfaces. As there is an extensive increase in the number of Web Services, the needs of automation for discovery, selection and composition of these Web Services have risen. In order to bring an automation task into a web service, a semantic description on the method of invoking a service, the way the service works, the order of calling a service and the functionalities it offers has to be added to the Web Services. There are two well-known Semantic Web Services Frameworks and these are the OWL-based web service (OWL-S) [1] and Web Service Modeling Ontology (WSMO) [2].

Many Semantic Web Services research works are focused on automation of discovery, selection and composition of the Web Services which are aided by ontology. Research findings have highlighted that the most challenging tasks during automatic discovery, selection and composition of the Semantic Web Services are diagnosing and resolving incompatibility between Web Services. As a result an important terminology "Mediation" in Semantic Web Services has emerged to handle incompatibility between Web Services.

Fensel and Bussler [3] have described mediation as "a *process for settling a dispute between two parties where a third one is employed whose task is try to find common ground that will resolve inconsistencies between their respective conceptualizations of a given domain*". Apart from the definitions of Fensel and Bussler, there are many other definitions for mediator in context of Web Services. For instance, Grahne and Kiricenko [4] define mediator as a "*software module that provides sharing of services and agglomeration of resources into complex service*".

There are three types of mediation, namely data, functional and process mediation. There are a significant number of researches on the Semantic Web Services that have explored data and functional mediation which is essential for automatic discovery, selection and composition. On the other hand process mediation has only been introduced as a supporting component in the composition of Web Services.

This paper focuses on process mediation in WSMO. For this study, current solutions are explored and open issues that are needed to be addressed and identified to support process mediation. It is clear that data mediation is an important element in process mediation and there is the dire need to mediate each incoming and outgoing data, before understanding the interaction between them. There are many existing researches on data mediation that support process mediation [5, 6] and therefore, the semantic or ontology in process mediation approaches are not mentioned in this paper.

The techniques identified have been used in understanding the interaction between the Web Services. It has been found that the existing process mediation approaches in the WSMO framework are tailored to a specific web service interaction scenario. There are many elements needing to be explored to support the automatic generic Web Service solutions.

Here on this paper is organized as follows: Section 2 describes the process mediation in WSMO Framework and provides definitions of process mediation and mediator components. It also explains how choreography interface supports process mediation. Section 3, describes process mediation approaches that uses the WSMO Framework. Section 4, provides a discussion on the current approaches and addresses the open issues that need to be explored to generate process mediator automatically for generic solutions. Finally, Section 5 provides some discussion and the conclusions.

II. Process Mediation in WSMO

This section describes the role of the process mediator in resolving mismatches in messages. It explains each type of the message mismatches and the ways to resolve them. All the important component of WSMO that play important roles in process mediation have been summarized as follows.

A. *Role of Process Mediator*

The process mediator is called the communication mediator in WSMO. Fensel and Bussler [3] identified three types of communication mismatches between the Web Services, namely precise match, solvable and irresolvable mismatches. A precise match occurs when the sender web service sends the message in the exact order that the receiver web service has requested. Therefore it only requires data mediation to solve possible data or format mismatches. The unsolvable message mismatches usually comes to a dead end. This paper focuses on the solvable mismatches that have been highlighted by many researchers. There are five situations that generate solvable message mismatches as stated below:

1) sender Web Service sends a message that is not expected by the receiver
2) sender sends single message that is expected to be in multiple forms
3) sender sends multiple messages that are expected to be in the single form
4) sender sends messages in wrong order
5) sender is not sending messages that are expected by the receiver.

Cimpian [7] has presented five ways that the process mediator could address solvable mismatches as listed below:

1) it stops the original message since it is not requested by the receiver
2) it splits the original message before reaching the receiver
3) it combines the original message before it reaches the receiver
4) it inverses the original messages before it reaches the receiver
5) It sends a dummy message since a message is expected by the receiver

Similarly, there are many researchers [8, 9] who have identified interaction patterns that are able to transform an original message into the required communication pattern.

B. *Mediator as WSMO Component*

This framework provides a rich description of all the related aspects of Web Services through four important components which are: the goal, web service, ontology and the mediator. The ontology component plays an important role in this framework since it carries the semantic description for all the other components in this framework. The goal component defines the user's preferences with respect to the requested functionality and interfaces through the *requestedCapability* and the *requestedInterface*. On the other hand, the web service component defines the offered functionalities and the ways to interact with the services through the capability and interface elements.

Generally, the goal, the web service and the ontology components play a common role to bring the semantic description to a Web Service which is similar to other Semantic Web Services Frameworks. However, this framework has proposed a distinctive component known as the mediator to resolve the interoperability problems in Web Services at various levels such as data, functional and process mismatches. This component contains four elements which are the *OOMediator*, the *GGMediator*, the *WGMediator* and the *WWMediator* to overcome interoperability problems between different the WSMO components.

Based on the definition provided in [2], the *WGMediator* and the *WWMediator* are closely related to the process mediator. However, the implementation of the *WWMediator* in resolving these process mismatches is not specified clearly in any of the provided example.

C. *Choreography Interface that Supports Process Mediation*

This section describes how process mediation takes place in the WSMO framework. The process mediation is closely related to the interface element of the goal and the web service components. The interface element describes how the functionality of a service can be obtained from two perspectives namely the choreography and orchestration interface.

The choreography interface explains communication methods between the service provider and the requestor whereas orchestration interface explains the communication methods among several Web Services. This paper limits the process mediation in the choreography interface due to limited resources available for process mediation in the orchestration interface. The two main elements in the choreography interface that supports process mediation are state signatures and transition rules. Figure 1 illustrates the elements in the choreography class which is extracted from [10].

Class choreography
hasNonFunctionalProperties **type** nonFunctionalProperties
hasStateSignature **type** stateSignature
hasTransitionRules **type** transitionRules

Figure 1. Choreography Interface

In WSMO, the choreography interface is described as using state-based technique which is based on Abstract State Machine (ASM) methodology. They state signature plays important role to define the mode or state of each instances it is used in the choreography interface. This state signature elements are described by an ontology in WSMO. Below are the five modes of state signatures as described in [10]:-

1) Static: any instance of relation and concepts in the "static" mode cannot be changed by both the provider and requester's choreography interface.

2) In: any instance of relation and concepts in the "in" mode can only be read by the choreography interface. It also means that the instance in "in" mode is expected as input by the choreography interface to invoke the service.

3) Out: any instance of relation and concepts in the "out" mode can only be created by the choreography interface. It also means that the instance that in "out" mode will be produced as an output during the invocation.

4) Shared: any instance of relation and concepts in the "shared" mode can be read and created by both the choreography interface.

5) Controlled: any instance of relation and concepts in the "controlled" mode can only be created and modified by the choreography interface.

These state signatures do not return the actual value of the instances during the invocation. They only contain a Boolean value (true or false). It returns true when an instance is required by the corresponding service such as instance with "in" and "shared" mode. It will also be stored in the internal repository since it can be useful in communication between the services.

The second important element in the choreography interface is the transition rules. They also termed as guarded transitions. A rule is triggered when the current state of the instance fulfills certain conditions. The rule does not reflect the actual system processing of the instance value. However, it expresses the data flow between the interacting Web Services.

III. Process Mediation Approaches using WSMO Framework

In this section, we will discuss four process mediation approaches that adopt the WSMO Framework and these are the message-based process mediation, the process mediation algorithm in Semantic Web Service (SWS) challenge, the process mediator as goal in IRS-III and the space based process mediation in Triple Space Computing (TCS).

A. Message based process mediation

In this WSMO Framework, the process mediation at runtime is as proposed in [11, 12]. It mediates communication mismatches between the provider and requestor by analyzing the state signature and transition rules in the goal and web service components.

Figure 2 illustrates process mediation in WSMO and the interaction between the choreography interfaces of the goal and web service. It also illustrates the role of the state signatures and transition rules in the process mediation and the main components in the process mediation such as the Choreography Parser, Internal Repository and the WSML Reasoner and Data Mediator.

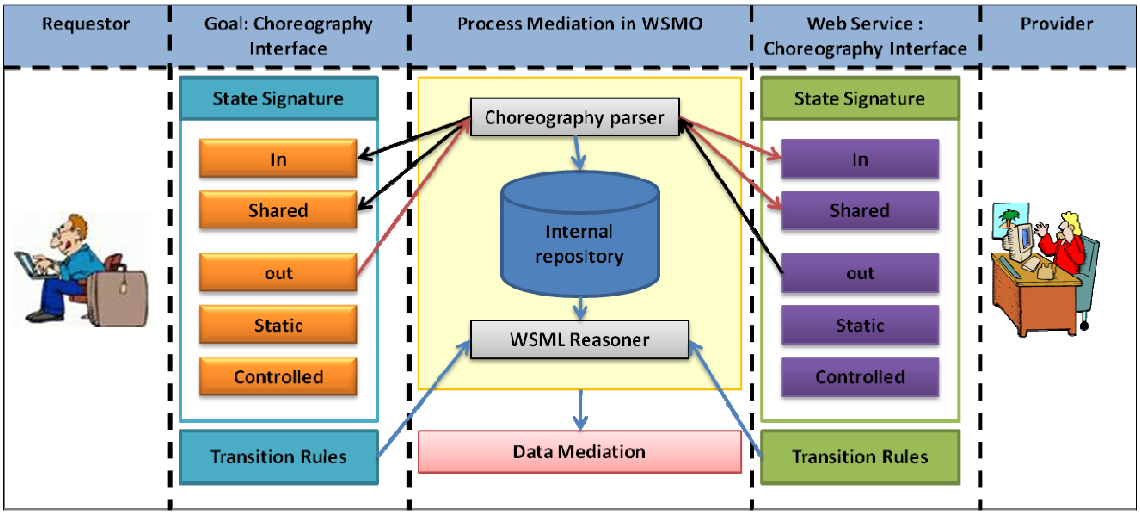


Figure 2. Message-based Process Mediation

This approach uses the Choreography Parser to check the mode of each state signature before it stores them in the Internal Repository. It matches the “in” mode instance in a goal/web service interface with the “out” mode instance from the web service/goal interface. This “in” and “out” list defines the data flow between the provider and the requestor. This data flow analysis is supported by the transition rule which defines the sequencing of the data exchange. This approach uses the WSML Reasoner to evaluate the transition rules before sending or deleting the stored instances.

However, this approach provides insufficient discussion on how choreography parser which matches the state signatures and the WSML Reasoner evaluates the transition rule that works together. Secondly, it also not utilizing the *WGMediator* or *WWMediator* components as explained in the WSMO concepts. Thirdly, it only provides steps to resolve process mismatches in a specific scenario but not for the general algorithm as to how the process mediator could perform the transformation based on the state signatures.

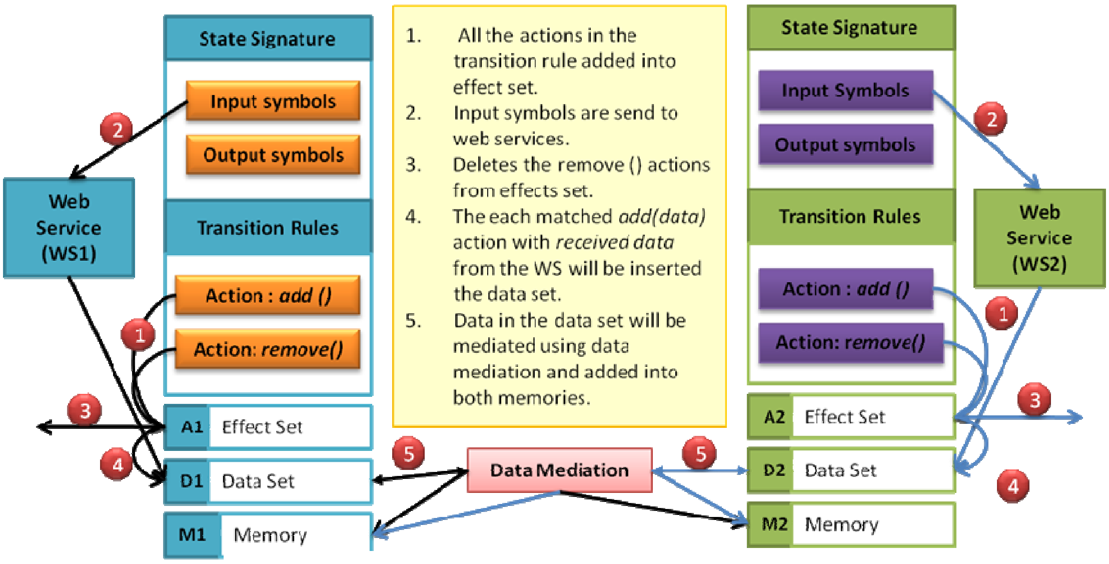
B. Process mediation algorithm in SWS Challenge

WSMO Framework has also extended to resolve the mediation scenario in the SWS challenge [13]. In this approach, the message interactions are evaluated using both the transition rules and the data flow that are extracted from the choreography interface. Two important contributions of this approach have been extracted in order to resolve the process mediation.

Firstly, it has defined four basic choreography rules that are derived from the WSDL operations. The WSDL operations are classified into four patterns; in-out, in-only, out-only, out-in based on the sequence of input or output messages of the operation. Secondly, this approach provides a general algorithm that handles the communication mismatches.

Generally, this algorithm collects all the data that needs to be exchanged between the Web Services and store them into the memory. Firstly, it evaluates the transition rules in each Web Services and stores required actions such as add and remove into the memory. At the same time, it also sends the input parameter in each web service as stated in the choreography interface.

Secondly, it evaluates the action list in the memory and deletes the removed action and the corresponding data from the data list in the memory. It then checks the output symbols at each web service. The output symbols that are equivalent to the messages in the add action will be inserted into data list in the memory and removes the corresponding add action from the action list. This algorithm ensures that the each web service memory contains the expected incoming messages. Finally, it calls the data mediation to mediate the data in both Web Service memories.



1. All the actions in the transition rule added into effect set.
2. Input symbols are send to web services.
3. Deletes the remove () actions from effects set.
4. The each matched *add(data)* action with *received data* from the WS will be inserted the data set.
5. Data in the data set will be mediated using data mediation and added into both memories.

Figure 3. Process Mediation in SWS Challenge

Figure 3 illustrates how the algorithm inserts the mediated data into the memory of the Web Services based on the choreography interface. This algorithm has addressed all mismatch patterns [7] except for the new generating messages. It allows message ordering and stops unexpected messages. The message merging and spliting is handled by the data mediation. This approach has provided a clear view on how the transition rules and the state signatures can be evaluated using an algorithm to generate the process mediation steps. Generally, this process mediation technique is similar to space based process mediator (SPM) [14] approach. It has provided memory space to the each web service to handle the data flow which is supported by the choreography transition rules. It does not specify the usage of the WSMO mediator components such as the *WGMediator* or *WWMediator* in the process mediation algorithm. Moreover, ability of the algorithm in handling complex mismatches or combination of more than one mismatch patterns are also underspecified.

C. *Process mediator as goal in IRS-III*

The main aim of Internet Reasoning Services (IRS) is to provide automated or semi-automated solutions for the semantically enhanced system over web. After few evolution of IRS, the IRS-III [15] has incorporated its existing framework which uses OCML, the ontology representation language with the WSMO core elements which are the goal, the web service and the mediator. This combination has produced a semantic broker-based approach that is able to mediate between the requester and the Web Services provider.

In IRS-III, the mediation task is resolved by the mediation handler component. This handler consists of the goal, the data and the process mediator. These mediators serve as a bridge between the semantic description such as the GG-Mediator, WW-Mediator, WG-Mediator and OO-Mediator with the other IRS components. The process mediator in this approach uses the GG and WW mediators to resolve four types of mismatches; a) not matched input/output, b) wrong order of input/output, c) output/input that needs to be split, and d) output/input that needs to be concatenated.

The process mediation in IRS-III is also closely related to the choreography and orchestration based service interactions. It also adopts ASM that contains states and transition rules to represent the interaction between the service provider and the requestor. However, it uses the forward-chaining-rule engine to execute the service interactions. In addition to the transition rules, this approach has defined choreography primitives to control the conversation between the IRS-III and Web Services.

Differing from WSMO/X approach, IRS-III does not load choreography interface of both goals of the requestor and web service provider. The IRS-III only evaluates the choreography interaction from the requester's perspective. Below are issues on process mediation in this approach.

- It has declares the mediators as goal which can be invoked as the mediation services. However, detailed explanation on how this mediation goal can be discovered and selected is based on the underspecified communication mismatches.
- The generation of the WW and GG Mediators supports the process mediator component is not specified.
- As for the other approaches, it also does not provide detailed description on reasoning mechanism used during evaluation on the transition rules.

D. *Space based process mediation in Triple Space Computing*

Apart from IRS-III, WSMO framework has also collaborated with Triple Space Computing (TSC) method to generate process mediation in the Semantic Web Services environment. TCS uses the Space-based Process Mediator (SPM) [14] approach to handle the communication mismatches between the service requestor and provider. The SPM method evaluates the data flow between the services using the data space. It also analyses the choreography rules that describe which data to be exchanged according to guarded rules through the control flow analysis.

TCS adopts the SPM method and provides a virtual data space, which is divided into the requestor and provider subspace. Generally, the TCS plays a middleware role between the requestor and provider, whereby all the sending and requesting messages take place through the TCS. It also handles process mediation by redirecting, transforming, stopping the data stored in the provider and requestor's sub space. The main feature of the TCS framework that supports the process mediation is the backend storage that provides shared virtual data space and the storage management mechanism that is able to store the history of interaction, monitor the interaction and resume interactions from point of failure.

In this approach [16], it has been described as to how the five resolvable message mismatches (as stated in WSMO) can be overcome by storing and transforming via the TCS virtual data space. However, the implementation of the actual WSMO concepts such as the goal, the web service, the mediator and the ontology in the TCS framework are underspecified.

IV. Discussion

The main aim of this paper is to identify the important component of process mediation based on the existing approaches that are related to the WSMO framework. The existing techniques can be discussed by two main perspectives; namely, identification of the mismatches and generation of the process mediator.

Firstly, for the identification of the communication mismatches, all the approaches are uses data analysis together with process flow analysis. The data flow is analyzed by comparing the expected messages of a web service with the actual incoming messages and outgoing messages with the expected messages in the web service of the recipient.

Generally, the data flow analysis techniques are supported by the transition rules which describe how the Web Services interact with each other. However the detailed description of the reasoning mechanism that is applied and the usage of ontology during the analysis of the transition rules are not available. In [13], data mediation is presented after identifying the data that is needed to be stored in the repository based on the choreography analysis in individual web services.

All these approaches however do not identify the five process mismatches specified in [3] according to the mismatch patterns. They only ensure that these mismatches are addressed in the approaches. Generally, a structured analysis on the data flow analysis which is supported by transition rules is still regarded as an open issue in order to address solution for generic process mediation.

Secondly, the techniques analyzed are involved in resolving the communication mismatches and that process mediators can be presented as web services [11, 12], algorithms [13], goals [17] and as virtual data space [16]. However, location, invocation and management issues of these process mediator solutions are also underspecified. The *WGMediator* and *WWMediator* elements that are related to the process mediation as stated in WSMO framework are not discussed in the actual implementation except for [17]. Figure 4 summarizes the open issues in semantic process mediation in the WSMO framework.

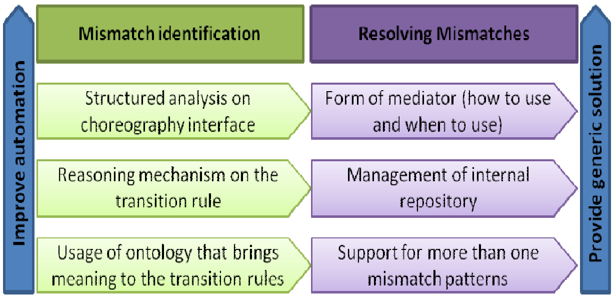

Improve automation
Mismatch identification
Resolving Mismatches
Provide generic solution
Structured analysis on choreography interface
Form of mediator (how to use and when to use)
Reasoning mechanism on the transition rule
Management of internal repository
Usage of ontology that brings meaning to the transition rules
Support for more than one mismatch patterns

Figure 4. Open Issues in Process Mediation

V. Conclusion

For this paper, the important elements of the process mediation in the Semantic Web Services are based on the analysis of the approaches that are been collaborated with the WSMO framework. WSMO components that specifically support process mediation are discussed. This followed by the process mediation approaches that are related to the WSMO framework. Based on the analysis, the similarities and difference between each technique of process mediation is identified and the common elements that support process mediation have been extracted from reviewed literature for this paper. The findings reveal that these elements can be extended to support the automatic communication mismatches in the generic Web Services.

Acknowledgment

This research is supported by Ministry of Science and Technology and Innovation (MOSTI), Malaysia and Universiti Teknologi Malaysia (UTM) under Fundamental Research Grant Scheme (FRGS) Vote Number 4F054.

References

[1] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara (2004), "OWL-S: Semantic Markup for Web Services ", W3C Member Submission, Retrieved from: http://www.w3.org/Submission/OWL-S/, Last Access: 28 June 2011

[2] J.d. Bruijn, C. Bussler, J. Domingue, D. Fensel, M. Hepp, U. Keller, M. Kifer, J. Kopecky, H. Lausen, E. Oren, A. Polleres, D. Roman, J. Scicluna, and M. Stollberg (2005), "Web Service Modeling Ontology (WSMO)", W3C Member Submission, Retrieved from: http://www.w3.org/Submission/WSMO/, Last Access: 28 June 2011

[3] D. Fensel, and C. Bussler, “The web service modeling framework WSMF,” *Electronic Commerce Research and Applications*, vol. 1, no. 2, 2002, pp. 113-137.

[4] G. Grahne, and V. Kiricenko (2005), "Process Mediation in Extended Roman Model", in M. Hepp, A. polleres, F. Harmelen, and M. Genesereth (Eds.) *First International Workshop on Mediation in Semantic Web Services (MEDIATE 2005) conjuction with 3rd International Conference on Service-Oriented Computing (ICSOC 2005)*, Amsterdam, Netherlands pp. 17-33.

[5] K. Gomadam, A. Ranabahu, Z. Wu, A.P. Sheth, and J. Miller, “A Declarative Approach using SAWSDL and Semantic Templates Towards Process Mediation,” *Semantic Web Services Challenge*, 2009, pp. 101-118.

[6] Z.X. Wu, K. Gomadam, A. Ranabahu, A.P. Shetb, and J.A. Miller (2007), "Automatic composition of semantic web services using process mediation", in J. Cardoso, J. Cordeiro, and J. Filipe (Eds.), LSDIS lab, University of Georgia pp. 453-461.

[7] E. Cimpian, and A. Mocan (2005), "WSMX process mediation based on choreographies", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Nancy, pp. 130-143.

[8] X. Li, Y. Fan, S. Madnick, and Q.Z. Sheng, “A pattern-based approach to protocol mediation for web services composition,” *Information and Software Technology*, vol. 52, no. 3, 2009, pp. 304-323.

[9] H.R. Motahari Nezhad, B. Benatallah, A. Martens, F. Curbera, and F. Casati (2007), "Semi-automated adaptation of service interactions", in *16th International World Wide Web Conference, WWW2007*, Banff, AB, pp. 993-1002.

[10] D. Fensel, H. Lausen, J.d. Bruijn, M. Stollberg, D. Roman, A. Polleres, and J. Domingue, “The Concepts of WSMO,” *Enabling Semantic Web Services*, 2007, pp. 63-81.

[11] E. Cimpian, “Message-based Semantic Process Mediation,” PhD Thesis, Faculty Science, National University of Ireland Galway, Ireland, 2010, p.198.

[12] E. Cimpian, and A. Mocan (2005), "Process Mediation in WSMX", in E. Cimpian (Ed.) Retrieved from: http://www.wsmo.org/TR/d13/d13.7/v0.1/, Last Access: 28 June 2011

[13] T. Vitvar, M. Zaremba, M. Moran, and A. Mocan, “Mediation using WSMO, WSML and WSMX,” *Semantic Web Services Challenge*, 2009, pp. 31-49.

[14] Z. Zhou, S. Bhiri, W. Gaaloul, and M. Hauswirth (2008), "Developing process mediator for supporting mediated Web service interactions", in *Proceedings of the 6th IEEE European Conference on Web Services, ECOWS'08*, Dublin, pp. 155-164.

[15] L. Cabral, J. Domingue, S. Galizia, A. Gugliotta, V. Tanasescu, C. Pedrinaci, and B. Norton (2006), "IRS-III: A broker for semantic web services based applications", in Athens, GA, United states, Vol. 4273 LNCS, pp. 201-214.

[16] Z. Zhou, B. Sapkota, E. Cimpian, D. Foxvog, L. Vasiliu, M. Hauswirth, and P. Yu (2008), "Process mediation based on triple space computing", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Shenyang, pp. 672-683.

[17] J. Domingue, L. Cabral, S. Galizia, V. Tanasescu, A. Gugliotta, B. Norton, and C. Pedrinaci, “IRS-III: A broker-based approach to semantic Web services,” *Web Semantics*, vol. 6, no. 2, 2008, pp. 109-132.

Context-Aware Services: A Survey on Current Proposals

Alfonso García de Prado
University of Cádiz
Cádiz, Spain
alfonso.garciaprado@mail.uca.es

Guadalupe Ortiz
Quercus Software Engineering Group
Mérida, Spain
gobellot@unex.es

Abstract— **Web services provide a successful way to communicate distributed applications, in a platform-independent and loosely coupled manner. Even though there are examples of good practice for the design, development and management of web services, there are scopes in which web service adaptation is required, such as context adaptation. Context-awareness is a complex topic to deal with but grants added value to any service which provides it. In this regard, there are multiple proposals in the recent literature which face the problem from different perspectives and using different technologies. This paper aims to show an overview of the most relevant approaches in this area, with a strong emphasis on those particularly related to the authors' work on the topic, namely using model-driven and/or aspect-oriented approaches.**

Keywords- Context-awareness, web services, model-driven development, aspect-oriented programming, context ontologies.

I. INTRODUCTION

Context and its use in software systems is a topic about which multiple research studies have been done in the last decade. This is not a surprising fact, since the design of applications and their communications whilst taking context into account permits the optimization of the use of information technologies in several respects: on the one hand, we can reduce the information submitted through communication lines to avoid their overhead; on the other hand; we will be able to save considerable resources in the client side and even in the servers', thanks to avoiding processing information which is not relevant for the device; finally, we will improve the user experience offering him a personalized service according to his requirements.

Context processing as well as adaptation to context is a hard task mainly due to the inherent complexity of the context itself and the multiple ways of managing it. In this regard, in this paper we will describe context state of the art, mainly focusing on those approaches which use a model-driven and/or aspect oriented development and on their usability for web services and those clients which access services from mobile devices.

We have focused our latest research on context adaptation for web services, specifically making the adaptation in the service-side and making it transparent for the client, which implementation would only have to provide the context information [1]. Our proposal is based on the use of aspect-oriented programming (AOP) for the decoupleness of the adaptation code and model-driven development (MDD) to simplify the system design without focusing on the final implementation or device requirements. We are currently extending this approach for other context issues and this is why we are specially interested in those research works where all these technologies interact in order to carry out context adaptation or context-awareness, although we will cover a wider area in this study of the state of the art.

In this sense, context adaptation in web services can be classified in two groups: first of all, adaptation in the service-side, where the process of transforming, selecting and adapting information depending on the client context is carried out in the service side. Secondly, client-side adaptation, which would in this case be the one in charge of following the mentioned process with all the information received from the service. Of course, these two options are not exclusive and there are some approaches which propose mixed models or approaches that, for instance, use a proxy or facade to develop the adaptation.

This paper is structured as follows: Section 2 introduces the definition and classifications of context. Section 3 provides an overview of recent research in the context scope, specially focusing on papers related to service adaptation and paying special attention to those which use model-driven or aspect-oriented techniques. Finally, Section 4 depicts a few conclusions in relation to the research works described.

II. CONTEXT BACKGROUND

Multiple definitions and discussions on the term context can be found ([2],[3], [4]), the one provided by Dey et al. in [5] being specially well-known –page 3, section 2.2: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*". One of the particular features of context information is that it is specific of each system, so that one specific type of information can be considered as part of the context in a particular system but not in a different one.

The term context-awareness supports the fact that the information provided about context by the client is properly used by the system so as to improve the quality of the interaction with it. That is, it means using information such as location, social attributes and other information from the user environment to foresee its necessities so that we can offer more personalized and easier to use systems. Therefore, a system is context-aware if it uses the context to provide relevant information or services to the user, adapting the system behavior to the particular needs of a specific user.

It is important to highlight that context information should be optional in context-aware applications, so that the application or service can still be delivered even if a lack of context information means it is not personalized.

It is difficult to establish a context classification since the term covers a wide range of topics, but we can distinguish three general types of context:

- Device-related context: it describes those features specific to system devices and communications among them, providing information about their current state (for instance *use*, *load*, etc.), capabilities and configurability. Such types of information, for example, consist of available networks and services, screen size and its orientation, available memory and battery, et cetera.
- Environmental context: it describes the environmental conditions in which user and devices are. Sensors are normally used in order to provide such kind of information as location, temperature, noise, et cetera.
- User context: users can specify their preferences in relation to configurable properties in their devices, be it personal data, office, hobbies, needs, et cetera.

Dealing with this context so that applications are aware of it implies distinguishing which part of the application is impacted. In this regard, we discern three different groups:

- First of all, the user interface; the way to represent the information and the way in which application and user can interact may vary depending on the context (if the device can represent images or not, if it is tactile, etc.).
- Secondly, the information itself may be affected by the context; for instance, if location is taken into account the result when searching a cinema would be restricted to the current city, or when checking the playbill from a mobile device, it would be better to avoid film trailers.
- Finally, we also have to take account of the changes in the functionality that a context can cause: for instance, if context is based on user preferences, when buying a cinema ticket online, one user's preference could be to pay online and obtain the ticket on a pdf file, but another user might wish to pay for and collect the ticket at the ticket office before the film starts.

Besides, we can also classify how we deal with the information in two different options:

- On the one hand, we find dealing with the amount of information depending on the context: for instance if the context information provides us with the screen size, when looking for information about running films, we may want to skip comments from other users, actor profiles, etc., for smaller screens.
- On the other hand, dealing with information content: in the same scenario, depending on user preferences we can provide information about one type of films or another.

Thus, we can assert that there is no doubt of the benefits of being able to adapt services to the context, yet the problem is the complexity of dealing with this adaptation.

III. Context-Awareness State of the Art

A. *Frameworks, middlewares and tools*

In general terms we can find some tools and middlewares which tend to support the development of context-aware applications and services, some of them mentioned in [6] and [7]. Yau and Karim [6] propose RCSM, a middleware with support for context-aware applications. RCSM provides a language for the specification of context requirements. RCSM obtains context real time data from different sources and provides them to objects which are analyzing the status. This system is for both conventional computers and PDAs.

There are also proposals specific to a particular programming language: Bardram provides a framework named JCAF which helps in the development of Java-based context-aware applications [9]. JCAF obtains information from context sensors and generates the appropriate classes to manage context in the final application.

PACE is a middleware proposed by Henricksen et al. [10] which provides support for user preferences and context. They also provide tools to facilitate the use of context information by the applications. PACE supports context based on user preferences.

SOCAM is a middleware presented by Gu et al. [11] which supports context logic and modeling based on OWL (Ontology Web Language).

The research pieces analysed so far are not specially designed to be integrated with web services, being more general approaches. In the following paragraphs we will summarize those which are specifically designed to deal with web services.

Keidl et al. introduce a framework which facilitates context adaptation in web service development [12]. Context information, provided through the SOAP message, can be processed in the service, the client or automatically by the dedicated framework.

The Akogrimo project presented by Osland et al. in [13] help mobile device users access and compute information in grid systems; it focuses on the location and environment context information, collecting the mentioned information and sending it through the context manager.

Chen et al. propose CA-SOA [14]; a context model is provided to describe the service and client-side context information. Based on the model, CA-SOA provides different components in order to facilitate discovery and access to context-aware services.

Anyserver platform, Han et al. [15], helps with context management for mobile services for varied context information such as device information and networks.

De Almeida et al. propose Omnipresent [16], a LBS (Location-Based Service) context adaptation system for web services. The context is modeled based on OWL; besides, different services to provide information based on location (such as maps or routes) are offered.

Truong et al. have several publications on the topic in question such as ESCAPE framework [17] and inContext Project [18], both designed for web services in workgroups and collaborative environments. They focus on emergency situations and provide techniques for modeling, storing and exchanging context information among web services.

Finally, CoWSAMI is a middleware proposed by Athanasopoulos et al. [19] to support context through the use of a *Context Manager* which deals with the different context sources, where context information can be queried.

B. *Context Ontologies*

Context ontologies deserve special attention; in this section we will examine some representative approaches:

Chen at al. propose CORBA-ONT [20], an ontology for the support of context-based systems expressed in OWL. It is a collection to describe locations, agents and events and their corresponding properties. A logic reasoning engine is also provided in order to deal with context information.

Korpipää et al. provides a context-based framework and ontology in [21], where a semantic definition is provided to manage multiple sources context information.

To end with, Ying and Fu-Yuan describe an ontology based on afore-mentioned SOCAM architecture, where the context is represented using OWL [22]. The proposed ontology focuses on intelligent home environments.

Even though context ontologies are a relevant topic, the approaches we find normally focus on very specific domains and do not solve the general problem of web service contexts. In contrast, W3C proposes Delivery Context Ontology [23], a more general approach providing a formal model for representing environment features for devices interacting on the Web. The proposed ontology includes, among other features, device characteristics and network used for connection.

C. *Client-Side and Proxy-Based Adaptations*

We can find several papers focused on the client-side or proxi-based adaptation; the following ones deserve special attention:

Laakko and Hiltunen present a content-based adaptation of information through a proxy [24]. They focus on converting XHTML (Extensible Hypertext Markup Language) into XHTML MP (XHTML mobile profile) and into WML (Wireless Markup Language).

URICA [25] is a technique for content adaptation for mobile devices. Mohomed et al. base this proposal on the system learning through its interaction with the user, identifying the most relevant context for the adaptation.

The previously mentioned paper from Korpipää et al. [21] focuses on the client-side context adaptation proposing a framework for mobile devices.

Lastly, Carton et al. propose a model-driven aspect-oriented development for generating context-aware applications for mobilephones ([26], [27]), although they do not deal with web services.

D. *MDD and Context Services*

Sheng et al. [28] propose a UML-based modeling language – ContextUML- for the model-driven development of context-aware web services. They show how UML can be used for dealing with context-information in a simple and flexible way. The proposed metamodel displays several classes:

a) First of all, a class identifying the context, which is extended by subtypes *AtomicContext* and *CompositeContext*: the first one represents a low level context and the second a higher level one which might be composed of other *Atomic* or *Composite* ones. For instance, temperature would be an atomic context, but weather would be composite.

b) Seconly, class *ContextSource* extended by *ContextService* and *ContextServiceCommunity* in a similar way as in the first topic.

c) The third class is *CAMechanism*, which is the one that formalizes context-awareness through two possible subtypes: *ContextBinding* and *ContextTriggering*; the latter would be formed by a set of *ContextConstrains* and another set of *Actions*.

d) Finally, class *CAObject* is the base class for any kind of element in a *ContextUML* model and shows four different subtypes, namely *Service*, *Operation*, *Message* and *Part*.

Sheng et al have presented a more recent paper in which a platform for the development of context-aware services is provided [29]. This platform, named ContextServ and based on ContextUML, provides an integrated environment where developers can specify and deploy context-aware services. The three main objectives of this platform are a) providing context definitions and facilities for specifying different context types; b) defining context-aware web services, for which purpose a graphic interface is provided; c) transforming the service model into the corresponding BPEL code.

E. *MDD, Aspects and Context-Awareness*

Carton et al. suggest, as previously mentioned, combining model-driven development and aspect-oriented programming [27] through the use of a set of tools: Eclipse Modelling Framework (EMF) is proposed for defining the metamodels. Theme/UML [30] is used for extending UML2.0 and OCL (Object Constraint Language) will allow developers to establish model restrictions. Then, Java Emitter Templates (JET) will permit the transformation of models into J2ME-based code. Finally, Graphical Modelling Framework (GMF) provides the way of defining models with a graphical representation.

An extension of the previous research is presented in [26] by Carton et al., where they provide a utility for model-driven transformations for mobile-based context-aware applications. They integrate Theme/UML through XMI (XML Metadata Interchange) using the UML editor MagicDraw. The openArchitectureWare tool is used for the generation of code from the XMI file.

Three phases are differentiated in their development process: first of all, Theme/Uml is used in the modeling stage, at the end of this phase and through the use of MagicDraw, they will obtain two files –Theme/UML MarkingProfileFile and UML2 Class Diagram File. Secondly, two transformations will be carried out at the composition phase: the input are the two files obtained from the modeling stage for the first one and a composition model is created from them; the second transformation uses the latter model to produce an EMF-based model with a platform independent oriented model. The final stage, through the use of XPand [31], consists of transforming the object-oriented platform-independent model into a platform-specific one, refining low level details until the final transformation of the platform-specific model into code.

F. MDD, Aspects and ContextServices

We can find some papers in which web services, model-driven development and aspect-oriented programming appear together:

Prezerakos et al. propose decoupling the main service functionality from the behavior related to the context using a model-driven development based on contextUML [31]. Thus, service business logic and context management are treated as separate issues at modeling and code level, where context code is encapsulated through an aspect-oriented implementation.

They modify the contextUML metamodel to utilize stereotypes with a view to: (1) reducing association between modeled services and context; (2) removing unnecessary relations; (3) simplifying the metamodel semantic in order to facilitate the model to code transformation.

Grassi and Sindico provide support for context adaptation in [33] by decoupling the adaptation process from the application business logic, their scope being service-oriented applications. For this purpose they define a framework based on model-driven and aspect-oriented software development (AOSD). Context and adaptation to context are modeled in separate sections.

As far as context modeling is concerned, the authors distinguish two types of context associated to entities: firstly, status-based context, which consists of a set of relevant attributes for the entity. An attribute in this context can be defined according to other attributes and can be associated to the source providing this information. Secondly, event-based context consists of a group of relevant events for the entity. In parallel, two types of constraint are defined as key elements for the introduction of context-awareness in the application, These are *state constraint* (defined by the logic predicate of the context value based on its state) and *event constraint* (defined as an event-based pattern).

Regarding the adaptation modeling, two mechanisms are provided to introduce context into the application: the first one (context-aware binding) is defined by a pair formed by an entity and a set of values, enabling the creation of different adaptation types depending on the entity type. The second one (context-aware insertion) is based on AOP. We can make two types of insertions (structural and behavioral) and in both cases the value to be inserted and location are provided. The structural insertion is equivalent to AOP intertype declaration using the location to specify the part of the application to be linked, the value specifying the elements to be injected. The behavioral insertion is equivalent to the advice concept in AOP, location being used to specify where to inject the new functionality (AOP join point) and the value to specify the new functionality itself.

Vale and Hammoudi [34] focus on the context-aware development of distributed applications proposing the use of model-driven engineering and separating matters of interest in different models. They focus on web service implementation and how to adapt them to changing contexts based on OMG EDOC-ECA [35] principles for context modeling and context-aware architectures. The result is CSOA, a *Context-aware Service Oriented Architecture*, based on the EDOC_ECA metamodel and providing business, context and composition views as platform-independent models and adaptation and service views as the platform-specific models, as we explain in the following paragraphs.

1. Business view provides traditional business logic.
2. Context view represents context information through the use of ontologies. A metamodel for context definition is also provided. Specifically W3C DRF (*Resouce Description Framework*) is used for the representation of context information at model level.
3. Composition view separates business from context logic in two different component types: business process components for the implementation of business logic, and contextual process components to provide the application's adaptation to context. The composition identifies which are the connections and interactions among all components.
4. Adaptation view provides a composition of one or more business and context components and a connection one. This composition is an abstraction of the process component and describes how process component instances are configured and connected for implementing the composition.
5. Service view is based on the WSDL metamodel. Services can be formed by other services represented in the context service composition.

Monfort and Hammoudi's proposal [36] shows two approaches to facilitate web service adaptation; the first one is based on an aspect-oriented implementation; the second one on the use of model-driven development for the context. For aspect-oriented implementation ASW (*Aspect Service Weaver*) is presented: it is a utility that lets us intercept SOAP messages between client and service adding new behaviors through the use of AOP, using XPath [37] for selecting WSDL methods to be intercepted. The second implementation represents a context metamodel identifying those issues considered more relevant for mobile devices. They also provide context parametrized transformations. Finally, they propose the combination of both techniques.

G. MDD, Aspects and Context –Awareness for Mobile Devices

We can also find some proposals which deal with context adaptation. In this line of work, Menkhaus presents an architecture for decoupling user interfaces in web applications from the application logic [37]. Dockhorn et al. propose an architecture to support mobile context-aware applications through a publish-subscribe mechanism [39]. None of these approaches deal with the adaptation of web service responses. Additionally, Pashtan et al. propose [40] the adaptation of web applications' content depending on the device, but do not tackle web service applications. On the other hand, Schomohl et al. provide context-aware mobile services [41], but they focus mainly on the creation of location-based services, rather than the adaptation of service responses to client requirements.

The work from Keith et al., whose framework was already mentioned [12], present an approach for services to deal with client contextual information through a context framework [42]. Context is always included in the client SOAP header as well as in service messages. This implies that not only services, but also clients have to process the context included in the header, however the proposal does not explore how the client can deal with the received context. In our proposal, the answer provided by the service is already adapted to client requirements, thus can be processed normally. Besides, their framework allows client context processing through the use of context plugins or context services. Context plugins have to be installed locally, which is improved by context services, available anywhere. A plugin and service have to be developed for each context and must be compatible with all services, which is extremely difficult and costly. Song et al. extend the latest work to preserve the client context privacy in [43], yet do not provide any further advantage to our proposal.

Concerning the client side, the proposal from Zhang et al. allows reengineering PC-based systems into a mobile product line by using a meta-programming technique. In their approach, systems are firstly developed for PC environments and then evolved to mobile device platforms by generating specific components from generic metacomponents [44]. The approach from Alves deals with existing variations in different mobile devices' models. He uses AOP to refactorize the variations and therefore decouple them from the core of the mobile application [45]. The idea of Blechsschmidt et al. is based on allowing the personalization of mobile device applications based on the end user profile [46]. For this purpose user information is collected and stored in XML files which are precompiled with the applications' core code in order for this information to be considered in the application during execution.

Finally, a relevant piece of work is the one presented by D. Zhang in [47]. He provides an approach for web content adaptation to meet user needs, suit characteristics of individual mobile devices, and adjust to dynamic contexts. His approach mainly focuses on web applications and an interactive adaptation in which users have to take part during the invocation to obtain the information they want. In the same line we can find the work from Niederhausen et.al. [48], where a framework for web applications adaptation is provided. The framework allows the developer to adapt web application content depending on different adaptation concerns, such as device adaptation, through the use of what they call adaptation aspects.

IV. CONCLUSIONS

Having studied the different approaches in related literature we are ready to depict a few conclusions:

Client-side adaptation does not remove unneeded information traffic and, even worse, overheads computation in the client side. In many cases, this computation becomes too complicated and even impossible to perform due to the excessive amount of unnecessary and useless data; for instance, if we wished to obtain home delivery restaurants in Cádiz (Cádiz being our context), a context-aware web service would remove those restaurants which are outside Cádiz from the initial list.

The use of a facade or proxy for the adaptation would imply the same problem. Client-side adaptation benefits from the advantage of not having to send the context to the service, but if we do it through a proxy then the mentioned information will be required. Finally, introducing a new element between service and client to manage the context creates a more complex architecture system, making its development and maintenance more difficult.

These significant drawbacks are solved when using service-side adaptation, in spite of having the inconvenience of having to send context information from the client to the service. For the adaption to context we consider that using model-driven development and aspect-oriented programming allows an easier implementation both at design and development phase, as well as at maintenance time. We also consider of special relevance the use of an ontology or other formalized classification and definition for context concepts. Based on these assumptions we plan to extend our previous work [1] for context adaptation as explained in [49].

ACKNOWLEDGMENT

The second author acknowledges the support from Ministerio de Ciencia e Innovación (TIN2008-02985) and Fondo Europeo de Desarrollo Regional (FEDER).

REFERENCES

[1] Ortiz, G. and Garcia de Prado, A. "Improving Device-Aware Web Services and their Mobile Clients through an Aspect-Oriented, Model-Driven Approach". Information and Software Technology Journal, Vol. 52, Issue 10, 2010, pp.1080-1093.

[2] Dey, A.K., "Understanding and Using Context", Personal and Ubiquitous Computing, vol. 5, 2001, pp. 4-7.

[3] Chen, G. and Kotz, D., A Survey of Context-Aware Mobile Computing Research, Hanover, NH, USA: Dartmouth College, 2000.

[4] Baldauf, M., Dustdar, S., and Rosenberg, F., "A survey on context-aware systems", International Journal of Ad Hoc and Ubiquitous Computing, vol. 2, 2007, pp. 263.

[5] Abowd, G.D., Dey, A.K., Brown, P.J. Davies, N. Smith, M. and Steggles, P. "Towards a Better Understanding of Context and Context-Awareness", Handheld and Ubiquitous Computing, vol. 1707, 1999, pp. 304-307.

[6] Truong, H.L and Dustdar, S. (2009) "A survey on context-aware web service systems", International Journal of Web Information Systems, Vol. 5 Iss: 1, pp.5 - 31

[7] Kapitsaki, G.M. Prezerakos, G.N. Tselikas, N.D. and Venieris I.S. "Context-aware service engineering: A survey", Journal of Systems and Software, vol. 82, no. 8, pp. 1285-1297, 2009.

[8] Yau, S.S. and Karim, F., "A context-sensitive middleware for dynamic integration of mobile devices with network infrastructures", Journal Parallel Distributed Computing, vol. 64, 2004, pp. 317.

[9] Bardram, J.E., "The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications", Pervasive Computing, Gellersen, H.W., Want, R., and Schmidt, A., Eds., Springer, 2005, pp. 98-115.

[10] Henricksen, K., Indulska, J., McFadden, T., and Balasubramaniam, S., “Middleware for Distributed Context-Aware Systems”, Int. Symp. on Distributed Objects and Applications, 2005, pp. 846–863.

[11] Gu, T., Pung, H.K., and Zhang, D.Q., “A service-oriented middleware for building context-aware services”, Journal of Network and Computer Applications, vol. 28, 2005, pp. 1-18.

[12] Keidl, M. and Kemper, A., “A Framework for Context-Aware Adaptable Web Services”, EDBT, 2004, pp. 826-829.

[13] Osland, P.O., Viken, B., Solsvik, F., Nyngreen, G., Wedvik, J., and Myklbust, S.E., “Enabling context-aware applications”, Proceedings of ICIN2006: Convergence in Services, Media and Networks, 2006.

[14] Chen, I., Yang, S. and Zhang, J., “Ubiquitous Provision of Context Aware Web Services”, 2006 IEEE Int. Conf. on Services Computing, Chicago, USA: 2006, pp. 60-68.

[15] Han, B., Jia, W., Shen, J. and Yuen, M.-C., “Context-Awareness in Mobile Web Services”, Parallel and Distributed Processing and Applications, Cao, J., Yang, L., Guo, M., and Lau, F., Eds., Springer Berlin / Heidelberg, 2005, pp. 519-528.

[16] De Almeida, D.R., Baptista, C.D., Da Silva, E.R., Campelo, C.E.C., De Figueiredo, H.F. and Lacerda, Y.A., “A Context-Aware System Based on Service-Oriented Architecture”, Int. Conf. on Advanced Information Networking and Applicationsm, V. 1, Vienna, Austria: 2006, pp. 205-210.

[17] Truong, H.-L., Juszczyk, L., Manzoor, A. and Dustdar, S., “ESCAPE – An Adaptive Framework for Managing and Providing Context Information in Emergency Situations”, Smart Sensing and Context, Kortuem, G., Finney, J., Lea, R., and Sundramoorthy, V., Eds., Springer Berlin / Heidelberg, 2007, pp. 207-222.

[18] Truong, H.-L., Dustdar, S., Baggio, D., Corlosquet, S., Dorn, C., Giuliani, G., et al., “inContext: A Pervasive and Collaborative Working Environment for Emerging Team Forms”, Int. Symposium on Applications and the Internet, Turku, Finland: 2008, pp. 118-125.

[19] Athanasopoulos, D., Zarras, A., Issarny, V., Pitoura, E. and Vassiliadis, P., “CoWSAMI: Interface-aware context gathering in ambient intelligence environments”, Pervasive and Mobile Computing, vol. 4, 2008, pp. 360-389.

[20] Chen, H., Finin, T. and Joshi, A., “An ontology for context-aware pervasive computing environments”, The Knowledge Engineering Review, vol. 18, 2003, pp. 197-207.

[21] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H. and Malm, E.-J., “Managing context information in mobile devices”, IEEE Pervasive Computing, vol. 2, 2003, pp. 42-51.

[22] Ying, X. and Fu-yuan, X., “Research on Context Modeling Based on Ontology”, 2006 Int. Conf on Computational Inteligence for Modelling Control and Automation and Int. Conf.on Intelligent Agents Web Technologies and International Commerce, Sydney, Australia: 2006, pp. 188-188.

[23] Cantera Fonseca, J.M. and Lewis, R., “W3C - Delivery Context Ontology”, http://www.w3.org/TR/dcontology/, 2009. Last access 06/2011

[24] Laakko, T. and Hiltunen, T., “Adapting Web Content to Mobile User Agents”, IEEE Internet Computing, vol. 9, 2005, pp. 46-53.

[25] Mohomed, I., Cai, J.C., Chavoshi, S. and de Lara, E., “Context-aware interactive content adaptation”, Int Conf on Mobile systems, applications and services , Uppsala, Sweden: 2006, pp. 42.

[26] Carton, A., Driver, C., Jackson, A. and Clarke, S., “Model-Driven Theme/UML”, Transactions on Aspect-Oriented Software Development VI, Katz, S., Ossher, H., France, R., and Jézéquel, J.-M., Eds., Springer Berlin / Heidelberg, 2009,p p. 238-266.

[27] Carton, A., Clarke, S., Senart, A. and Cahill, V., “Aspect-Oriented Model-Driven Development for Mobile Context-Aware Computing”, Int. W. on Software Engineering for Pervasive Computing Applications, Systems and Environments, USA, 2007, pp. 5-5.

[28] Sheng, Q.Z. and Benatallah, B., “ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development”, International Conference on Mobile Business, Sydney, Australia: 2005, pp. 206-212.

[29] Sheng, Q.Z., Pohlenz, S., Yu, J., Wong, H.S., Ngu, A.H.H. and Maamar, Z., “ContextServ: A platform for rapid and flexible development of context-aware Web services”, Int. Conf. on Software Engineering, Vancouver, BC, Canada: 2009, pp. 619-622.

[30] Clarke, S. and Baniassad, E., Aspect-Oriented Analysis and Design: The Theme Approach, Addison-Wesley Professional, 2005.

[31] XPand. http://www.eclipse.org/modeling/m2t/?project=xpand, [Last access: August 2011

[32] Prezerakos, G.N., Tselikas, N.D. and Cortese, G., “Model-driven Composition of Context-aware Web Services Using ContextUML and Aspects”, IEEE Int. Conf. on Web Services, Salt Lake City, USA: 2007,pp. 320-329.

[33] Grassi, V. and Sindico, A., “Towards model driven design of service-based context-aware applications”, Int. W on Engineering of software services for pervasive environments, Dubrovnik, Croatia, 2007, pp. 69-74.

[34] Vale, S. and Hammoudi, S., “Model Driven Development of Context-aware Service Oriented Architecture”, Int. Conf. on Computational Science and Engineering - Workshops, San Paulo, Brazil, 2008, pp. 412-418.

[35] OMG EDOC-ECA. http://www.omg.org/spec/EDOC/1.0/ [Last access: August 2011]

[36] Monfort, V. and Hammoudi, S., “Towards Adaptable SOA: Model Driven Development, Context and Aspect”, International Joint Conference on Service-Oriented Computing, 2009, pp. 175 - 189.

[37] Xpath. http://www.w3.org/TR/xpath/ [Last access: August 2011]

[38] Menkhaus, G. “Architecture for client-independent Web-based applications”. Proceedings of the Technology of Object-Oriented Languages and Systems, Zurich , 2001, pp. 32-40.

[39] Dockhorn Costa, P., Ferreira Pires, L., van Sinderen, M. and Pereira Filho, J., “Towards a Service Platform for Mobile Context-Aware Applications”. Int. W. on Ubiquitous Computing, Portugal, 2004, pp.48-61.

[40] Pashtan A., Kollipara S. and Pearce, M. “Adapting Content for Wireless Web Services”, IEEE Internet Computing, V. 7, N. 5, 2003, pp. 79-85.

[41] Schomohl, R. and Baumgarten, U. “Mobile Services based on client-server or P2P architectures facing issues of context-awareness and heterogeneous environments”. Int. Conf. on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 2007.

[42] Keidl, M. and Kemper, A. “Towards Context-Aware Adaptable Web Services”.Int. WWW Conference on Alternate, New York, 2004, pp.55-65.

[43] Song, Y. J., Lee, D.H., Yim, J.G. and Nam, T. Y. “Privacy Aware Adaptable Web Services Using Petri Nets”. Int.Conf. on Convergence Information Technology, Korea (South), 2007, pp. 19-24.

[44] Zhang, W., Jarzabek, S. Loughran, N. and Rashid, A. “Reengineering a PC-based System into the Mobile Device product Line”. Int. W. on Principles of Software Evolution, Helsinki, Finland, 2003, pp.149-160.

[45] Alves, B. “Identifying Variations in Mobile Devices”. Journal of Object technology, Vol 4, Nº 3, 2004, 47-52.

[46] Blechsschmidt, T., Wieland, T., Kuhmunch C. and Mehrmann, L.. “Personalization of End User Software on Mobile Devices”. Int. W. on Mobile Commerce and Services, München, Germany, 2005, 130-137.

[47] Zhang, D. “Web Content Adapatation for Mobile Handheld Devices”. Communications of the ACM, Volume 50, Issue 2, February 2007, pp. 75-79.

[48] Niederhausen, M., Fiala, Z., Kopcsek, N. and Meissner, K., “Web Software Evolution by Aspect-oriented Adaptation Engineering”, IEEE Int. W. on Web Site Evolution, Paris, France: 2007, pp. 3-7.

[49] Ortiz, G. and Garcia de Prado, A. “Web Service Adaptation: A unified approach versus multiple methodologies for different scenarios”, ICIW 2010, the International Conference. on Internet and Web Applications and Services, Barcelona, Spain, 2010, pp.569-572.

Semantic Web-driven Agent-based Ecosystem for Linked Data and Services

Oleksiy Khriyenko
Industrial Ontologies Group, MIT Department
University of Jyväskylä, P.O. Box 35(Agora)
Jyväskylä, Finland
oleksiy.khriyenko@jyu.fi

Michal Nagy
Industrial Ontologies Group, MIT Department
University of Jyväskylä, P.O. Box 35(Agora)
Jyväskylä, Finland
michal.nagy@jyu.fi

***Abstract* — We are surrounded by data – data about events, our daily activities, a multitude of products and services from different vendors, etc. This data is playing a central role in our lives. It helps us make better decisions. Increasing numbers of individuals and organizations are contributing to this huge flow by sharing their data with others, including Web-native companies (such as Google, Amazon, Facebook, YouTube, Twitter, etc.), newspapers, public governmental bodies, various research initiatives, etc. In turn, third parties are consuming this data to build new businesses, provide new services and accelerate scientific progress. However, very often new service creation has an obstacle – limited data availability. Becoming accessible later, data may cause reimplementation of the service that might cost too much and user will be left without improvement of the service. In this paper we combine the existing technologies, highlight the challenges and show the way that might help solve the problem. In order to elaborate Semantic Web-driven Agent-based Ecosystem for Linked Data and Services we utilize the so-called UBIWARE platform. UBIWARE is a semantic middleware platform for Ubiquitous Computing.**

Keywords – linked data, semantic service ecosystem, agent technology, service infrastructure, semantic service integration

I. INTRODUCTION

In the beginning of the computing era, simplicity of programming languages allowed people to write programs on paper. However, the complexity and size of programs has grown. Also, large programs were thought of as complex interconnected graphs in comparison with the linear structure of text document. As a result we had to make more efforts to express the full program as a collection of text files. Programs were separated to different files depending on the nature of data and role it played. Previously, programs were concentrated on local data, which was used from the memory of the computers that were running it. The majority of programs today have to do both: persist data and connect to remote data that is managed by another party.

In order to increase their competitiveness, many organizations are looking for a way to enhance their internal data with external information. However, the integration of information from different heterogeneous systems is challenging. In order to achieve it, we have to concentrate our efforts not only on an internal system integration framework, but also on a common approach towards distributed collaborative environment of heterogeneous components. The main trend of programming today is that most new system will not only need to persist their data, but will also need to connect to other programs across the Internet. Indeed, we often do not know which data will be local or which will be remote. Thus, now it is an appropriate time for systems and services to treat all data references as potentially pointing to data that resides somewhere else on the Internet. By taking this homogeneous approach, programmers can focus on the basic logic of what needs to happen rather than care about the data. Building connections between systems currently requires extra work that has to be repeated for each connection. This is additional work from a technical and financial point of view. Rather than building huge data storages, we should think about data as a complex graph that is connected across and distributed over many systems. The service oriented architecture (SOA) [1] approach to connecting existing data sources is an important step towards making it easier for existing systems to communicate with each other. However, again, in contemporary programming languages, the SOA approach has to be managed explicitly by the programmer. If we adopt programming paradigms that treat all data as being data on another system, we free the programmer of certain routine tasks and giving him/her more time to concentrate on other aspects of the system. Instead of huge data storages we should think about networks of data.

To achieve the vision of ubiquitous Web, the next generation of integration systems will need different methods and techniques such as Semantic Web [2] [3], Web Services [4] [5], Agent Technologies [6] and Mobility [7] [8]. Semantic technologies are viewed today as a key technology to resolve the problems of interoperability and integration within the heterogeneous world of ubiquitously interconnected objects and systems. It is evident that for two systems to communicate with each other, they have to use a standard language that they can both understand and share a standard ontology. There are different points of view concerning the uniqueness of the ontology. Some think that "one ontology approach" is the best possible solution to have one common standard and avoid ambiguity. Others consider this as an illusion and, rather than dreaming of agreeing on one common ontology, they suggest always working on the assumption that our systems should handle interactions with other systems using multiple different ontologies even within a single domain. It is one more aspect that increases the complexity of data management and distributed system interoperability.

The rise of software development methodologies further separated the way that programs and data are managed. There are many modern software systems with a lot of configuration options that can be user customized. It removes a clear boundary between the programming and pure data management. Nowadays, users want even greater control over how their computers manage their data and would like to become programmers on at least a higher system level. Users would like to manage not only the data, but program a system behavior through its configuration.

All the mentioned difficulties and complexity of the upcoming Web (web of autonomous intelligent systems and web of data) tell us about the necessity to elaborate the correspondent ecosystem with the appropriate tools and capabilities to support each player in the Web to operate there in a proper way. The Ecosystem should hide all the complexity and control the technical part of interoperability and unambiguousness. This paper presents a possible way of ecosystem elaboration based on agent-driven infrastructure for Linked Data and Services interoperability.

In the second section, we propose an architecture that would simplify the creation of ubiquitous services. We stress the importance of the Linked Data approach as an important enabler of service interoperability. The third section describes the infrastructure of such an ecosystem based on Multi-Agent technologies. In the fourth section we address several challenges of the Linked Data infrastructure. The fifth section concludes the article.

II. Ecosystem for Linked Data and Services

A. Towards Data-free Ubiquitous Services

In today’s world we can find different kinds of data in different forms. In general this data can be utilized by third parties in order to provide additional services. However, very often new service creation faces an obstacle in the form of data unavailability. Usually, ignorance of certain data availability and accessibility limits us to develop one or another useful service. If some data becomes available and accessible later and some existing service would like to utilize it, it may cause a service reimplementation. This reimplementation might cost the service provider too much in terms of time and money. Eventually, the user might be left out without any service improvement. To avoid such a case and allow new services to be developed in a more flexible way, we have to consider Semantic Web-based infrastructure for linked data and services. Standardized approaches like Semantic Web technology and Ontology-based development may help us to develop services, which are not bound to data, but operate with data on the semantic level. Following this approach, the service logic can be independent of the particular data source availability and still provide a service to the customer. Such a service elaboration approach and semantic organization of virtual data source makes the servicing context-aware, less sensitive to the unavailability of data and open/extendable for data that might be accessible in the future. Thus, one of the main parts of service infrastructure is common shared virtual data source – Semantic Data Space of linked information. Fig. 1 shows a general architecture of manageable Linked Data infrastructure for services and applications.

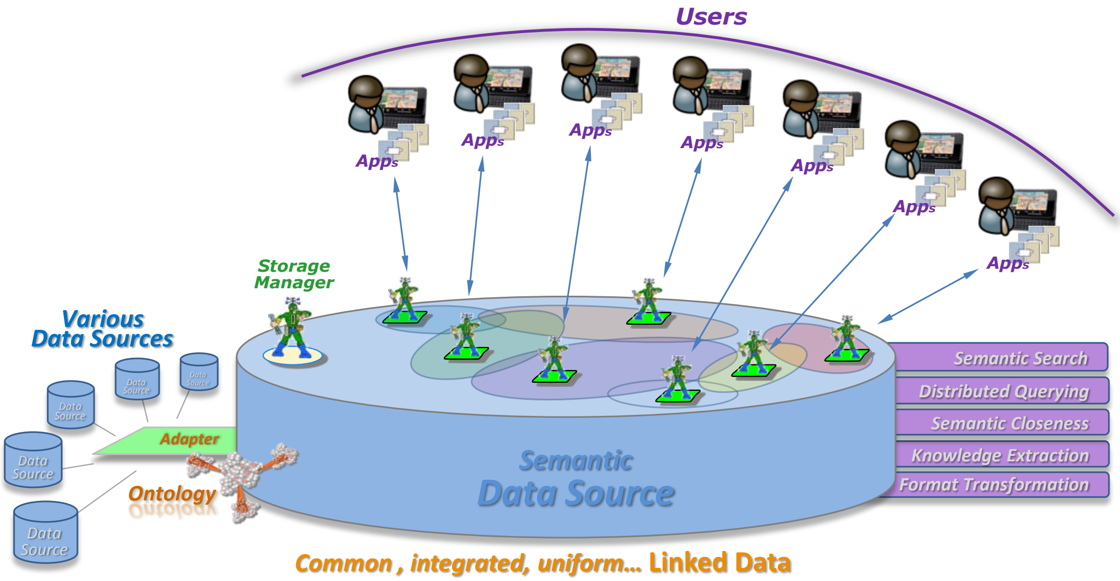


Figure 1. Semantic Data Space of linked information.

This smart data storage is an intelligent mashup of different heterogeneous sources of information with a dynamic private space for each user. It might be a source of any public or private information with correspondent access control mechanism. Depending on the user profile, activity performed by him/her and contextual information (e.g. location), the smart data storage should create the correspondent information space for services and applications used by user for the mentioned activity. In other words, this space will contain only the activity- and context-related linked data. Each information space has a manager that manages the data, adds relevant data and removes irrelevant data from the space. Such smart semantic organization of the common data storage allows services/applications to be automatically switched between different real data sources on-the-fly depending on the context. Such data organization allows creation of mashups through flexible semantic orchestration and user-driven choreography of services and applications. Such an approach very well supports and complements the new strategy of Nokia towards Internet-based mobile application solutions and the mixed reality concept. Following this approach, applications will get access to application independent, but contextually relevant, information. Such architecture allows us to move the data accessing and data processing part to the application independent layer and perform a reliable and trusted data access control.

For example, let us consider some notification service in the user’s mobile phone, where the user has specified his/her interest in “yacht exhibition” events. The user is not going to visit some specific exhibition that might take place somewhere far away, but he/she would be interested in visiting an exhibition close to his/her current location. In this case, the contextual information is the user’s location and the type of relevant events. Based on this context, the ecosystem creates a semantic data space with relevant information published by other systems/services (exhibition centers, yacht clubs, city and region event centers, etc.). The application in the mobile phone is developed in a general way and does not care about the data. It just sends the appropriate pattern of the request and listens to the response from the smart data space manager, which performs all search and matching processes. Even in the case when a certain exhibition has a restriction and is open only for the members of some yacht club, the application does not care about this fact. The correspondent data space manager itself checks this information based on the user profile and available linked data about the yacht club members. Thus, the complete semantic data space takes the responsibility for linked data management and allows applications and services to become ubiquitous and data independent.

B. Linked Data as a Basis for Service Interoperability

Unstructured data, heterogeneity of different data sources and many other problems become a bottleneck of automated data integration, processing and reuse. To make data ready to be processed by external intelligent algorithms and methods, data sources and data should pass through the semantic adaptation. If we are aiming at making a step towards intelligent data processing, to intelligent use of information from different data sources, to intelligent management of huge data storages, to knowledge extracting from huge archives of data, etc., we have to make the data linked. The data should be accessible in common uniform way through one virtual linked semantic data source, even if originally it is located in different data sources.

From the early beginning, the World Wide Web (WWW) was a system of interlinked hypertext documents accessed via the Internet. The Web allows document creators to freely choose whether they will or will not refer to any other document. As a result we have got a huge mass of information that was managed by search engines and browsers. However, with rapidly growing amount of information on the Web, the society needed some advanced mechanisms for data management. Later on, Semantic Web technology was announced as a "web of data" that enables machines to understand the semantics (meaning) of the information on the WWW. It extends the network of hyperlinked human-readable web pages by inserting machine-readable metadata about pages and how they are related to each other. This enabled automated agents to access the Web more intelligently and perform tasks on behalf of the users. It was the first attempt to arrange and standardize the data and data management.

Due to huge amount of areas and aspects that Semantic Web technology tried to cover, the community started to elaborate different standards and techniques to solve different problems. As a result, we have a big variety of separated islands of information and management systems. These information islands internally follow the Semantic Web vision, but are heterogeneous from the general (global) interoperability point of view. This leads to the fact, that society and especially its business-oriented part started to doubt that such widely spread activity will be so much beneficial for them. Only some applications and systems in restricted domains became really useful. Most probably, the reason for this is the decentralization of uncontrolled activities, which creates new problems and bottlenecks on the way towards ubiquitous Semantic Web.

It was evident that sooner or later the concepts and ideas of Semantic Web will be reformulated and presented in a simpler way to show a small but important step of technology applicability. As we see nowadays, the Linked Data concept, introduced and promoted by the fathers of WWW and Semantic Web [9] [10], becomes popular. The concept describes a method of publishing structured data, so that it can be interlinked and become more useful. Just as the WWW has revolutionized the way we connect and consume documents, Linked Data can revolutionize the way we discover, access, integrate and use data. It is built upon the Web as the ideal medium with ubiquity, distributed and scalable nature, mature and well-understood technology stack. There are no doubts that Semantic Web is a very promising technology of the future. It definitely lacks more centralized management or at least an environment that plays coordinative and supportive role and directs users to the proper utilization of the technology.

According to Tim Berners-Lee’s “5 stars” advice to enable Linked Data [9], the principles include: making data available on the Web (in whatever non-proprietary format) as machine-readable structured data, utilizing open standards from W3C (RDF [11] and SPARQL [12]) to identify things and finally linking the data to other people’s data to provide the context. But, it seems that it is not enough to define only requirements or principles. To facilitate proper utilization of the technology and increase the benefits of it, it is reasonable also to provide technical support in a form of Semantic Web oriented ecosystem platform with appropriate tools and services that do all the “dirty work” and keep the whole ecosystem in proper condition.

III. AGENT-BASED ECOSYSTEM INFRASTRUCTURE

Semantic Data Space is a complex data management system based on autonomous data space managers, distributed original data sources and supportive ecosystem tools and capabilities. In recent years, complexity of computing environments has grown beyond the limits of human system administrators’ management capabilities. With the advent of service-oriented computing (SOC), computing environments have become open and distributed, and components are no longer under a single organization control. Autonomic computing systems are expected to free system administrators to focus on higher-level goals [13]. Self-configuration, self-healing and self-optimization can be performed by autonomic computing systems without human intervention. As such, autonomic computing systems strongly resemble multi-agent systems (MAS). MAS, in turn, interact with services, as designed and developed within SOC. To make the system intelligent, dynamic and autonomous, we have to utilize Agent Technologies [14]. From the implementation point of view, agents are the next step in the evolution of software engineering approaches and programming languages. It is a step following the trend towards increasing degrees of localization and encapsulation in the basic building blocks of the programming models [15].

Developing and maintaining large-scale, distributed applications is a complex task. Middleware has traditionally been used to simplify the application development by hiding low-level details and by offering generic services that can be reused and configured by application developers. However, the middleware technology has not been keeping up with the growing demands that emerge in the digital society. The scale of distributed applications is rapidly increasing. The range of users that compose and configure applications has expanded significantly, and the increased scope of distributed applications has also resulted in more advanced application composition scenarios.

A. UBIWARE Platform: Integration Infrastructure for Heterogeneous Distributed Components

We base our research towards the elaboration of Semantic Web – driven Ecosystem for Linked Data and Services on UBIWARE platform that follows the GUN vision [16]. The Platform is a development framework for creating multi-agent systems. It is built on top of the Java Agent Development Framework JADE [17], a Java implementation of IEEE FIPA specifications. The name of the platform comes from the name of the research project, where it was developed. The UBIWARE project [18] introduced a new paradigm in software engineering and elaborated an approach towards the creation of semantically enhanced agent-based integration middleware that makes heterogeneous resources proactive, goal-driven and able to interoperate with each other in collaborative environment. In this project, a multi-agent system was seen, first of all, as a middleware providing interoperability of heterogeneous resources and making them proactive and “smart”.

The core of the platform gives every resource a possibility to be smart by connecting a software agent to it. This agent enables the component to proactively sense, monitor and control its own state and communicate with other components. The core component of the UBIWARE platform is a UBIWARE agent depicted in Fig. 2. It can be seen as consisting of three layers: the Behavior Engine implemented in Java, a declarative middle-layer (Behavior Models corresponding to different roles the agent plays), and a set of sensors and actuators, which are again Java components. We will refer to these as Reusable Atomic Behaviors (RABs). The middle layer is the beliefs storage presented in *Semantic Agent Programming Language (S-APL)* [19], which is a Resource Description Framework (RDF) [11] - based language. S-APL integrates features of agent programming languages (like AgentSpeak [20] and AFAPL [21]), semantic reasoners (like CWM), querying languages (like SPARQL [12]) and agent communication content languages (as FIPA SL [22]).

The main goal of the Platform is to provide interoperability between heterogeneous resources (applications and systems in our case). This can be achieved by semantic adaptation and by assigning a proactive agent to each of the resources. The communication, resource discovery and resource usage are performed via the correspondent agent. The Platform has inter-platform communication mechanisms and allows integration, orchestration and choreography of resources registered and located in different platforms. UBIWARE is not an application like an operating system, word processing software or Internet browser. It is a set of tools that helps people develop software. With respect to Cloud-based integration environment interoperability, we consider the UBIWARE platform a tool that allows automatic discovery, orchestration, choreography, invocation and execution of different Business Intelligence services.

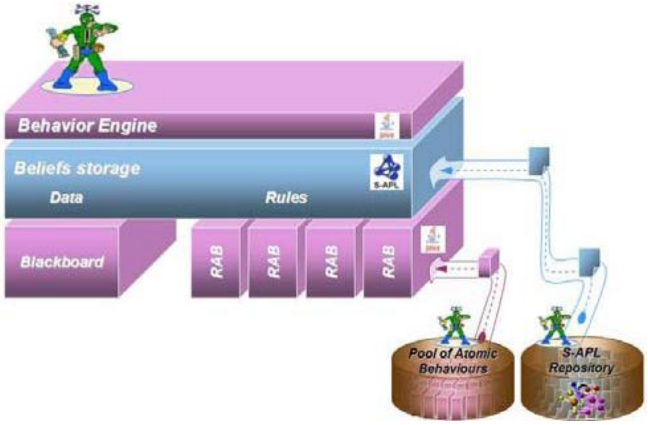


Figure 2. UBIWARE Agent architecture.

B. Semantic Data Space as a complex data management system

The UBIWARE platform-based ecosystem assumes that we have a network of interconnected platforms that can be extended any time it is needed. As it was mentioned in the previous section, the core entity of the platform is an agent. Each resource (data source, service, human, etc.) has an agent assigned to it. The agent represents its resource and performs all the communications with other resources (resource agents) in the ecosystem.

Users, data source adapters and other applications manually, semi-automatically or automatically create and edit the Web of linked data. The first layer of the ecosystem (see Fig. 3) is populated by Platforms that help to create, store and manage this data. On the same layer, the ecosystem has supporting tools, interfaces and other infrastructure services (browsers, search engines, similarity measurement modules, various supportive registries and systems) to be utilized by Platforms' managers and users. Thus, this layer represents a network of linked data.

Another layer is represented by agents. These agents are managers of personal data spaces. They form a smart semantic data space of the ecosystem. Such personal data spaces are generated each time when some application or service starts to consume linked data. These managers are responsible for proper context-dependent data access control and for relevance and unambiguousness of provided data. In other words, based on contextual data provided by applications/services, they on-the-fly create and manage datasets that perfectly match correspondent patterns provided by applications. During the whole life-cycle of the data space, the manager agent updates, adds, deletes the correspondent dataset with respect to the changes of the contextual information (user profile, location, tasks and conditions, etc.). To enable this, the manager communicates with the application and with other platform managers. It also utilizes infrastructure services to browse, search and filter linked data.

C. Capabilities and Tools of the Linked Data Infrastructure

The main functionality and main purpose of the proposed ecosystem is to provide infrastructure and tools that facilitate the process of Linked Data creation, browsing, search and access. Such infrastructure can be considered a perfect playground for semantic-driven applications and services that are given ubiquitous access to the complete data space.

A key factor in the re-usability of data is the extent to which it is well structured. The more regular and well-defined the data structure is, the easier it is to create tools to reliably process it for reuse. While most Web sites have some degree of structure, the language in which they are created, HTML, is oriented towards structuring textual documents rather than data. As data is intertwined with the surrounding text, it is hard for software applications to extract snippets of structured data from HTML pages. There were attempts to use various microformats to embed data to HTML, but all of them are restricted with a small set of types of entities and attributes and are not suitable for sharing arbitrary data on the Web. A more generic approach to make structured data available on the Web is Web API usage. Web APIs provide simple query access to structured data over the HTTP protocol. The advent of Web APIs has led to an explosion in small, specialized applications (or mashups) that combine data from several sources, which are accessed through different APIs. While the benefits of programmatic access to structured data are indisputable, the existence of a specialized API for each data set requires significant efforts to integrate each novel data set into an application. Every programmer must understand the methods available to retrieve the data from each API, and write custom code for accessing data from each data source. Thus, Web APIs make data accessible on the Web, but they do not place it truly on the Web, making it linkable and therefore discoverable.

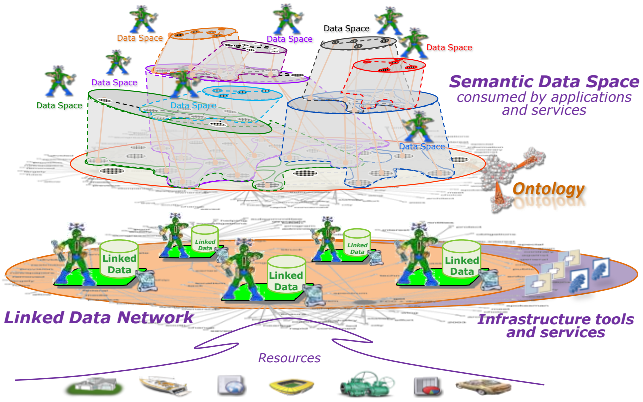


Figure 3. Semantic Data Space – a complex data management system.

The GUI of the Platform should help the user create Linked Data in the appropriate form. This controllable way of Linked Data creation minimizes the user's efforts and hides all technical complexity of the process. Such GUI has to be smart and configurable. This allows the user to connect and utilize different domain specific ontologies, and provide context dependent guidance for the Linked Data creator. At the same time, together with simple static resource transformation, we have to consider more complex resources and systems that produce dynamic data. There are a lot of huge data storages (that cannot be managed manually) and systems that produce an avalanche of data. This data is very often available only in human readable form. Within the UBIWARE platform we elaborated a resource adaptation framework and tools that facilitate creation of semantic adapters for various resources. These tools can automate the adaptation process for huge and dynamic data sources. It is possible to automate the process of Linked Data creation via correspondent data adapters. As a proof of this approach, in the iSCOPE project we adapted several web pages and services that dynamically publish event data. Then, we semantically annotated this data and made it available to remote mobile applications.

Nowadays, the human becomes a dynamic and proactive player in a large heterogeneous distributed environment with a huge amount of various data, services, devices, etc. Therefore it is necessary to provide a technology and tools for easy and handy human information access and manipulation. If the Web of Data is based on standards and a common data model, it will become possible to implement generic applications that operate over the complete data space: Linked Data browsers (which enable the user to view data from different data sources), Linked Data Search engines (that crawl the Web of Data and provide sophisticated query capabilities on top of the complete data space). These two general types of applications should be considered as general infrastructure tools/capabilities and used by all other applications as services in the Linked Data ecosystem. But, such functionality should not only be available for humans. The Ecosystem infrastructure should provide the same capabilities for machines or applications as well. This means that the Platform should be equipped not only with GUIs and tools for human operation on Linked Data, but also equipped with APIs, correspondent GUIs and utilities for application and service developers.

Another important aspect related to data browsing is context-sensitive visualization of data to a user [23] [24]. Context-awareness and intelligence of the user interface bring a new feature that gives the user a possibility to not get just raw data, but also required information based on a specified context. A user needs a fast and convenient way to specify what he/she is looking for and get the semantically closest resources to his/her query. Resource closeness/similarity search is one of the most useful features that users need during the information retrieving process [28]. The similarity search has become a fundamental computational task in many applications. Thus, intelligent visualization of Linked Data and context-aware filtering of relevant data becomes a very important functionality of the GUI and Linked Data browsers.

IV. Challenges of Linked Data Infrastructure

A. *URI aliases vs. Semantic Web ecosystem control*

One of the bottlenecks of Semantic Web is the huge amount of URI aliases, the multitude of URIs in different namespaces identifying the same entity. There is a common agreement saying that two URIs referring to the same resource should be connected using a link of type “http://www.w3.org/2001/07/owl#sameAs”. However, due to lack of control, these links are not used often. As a result, in many cases there are several unlinked information sources that describe the same resource. At the same time, even if the “owl#sameAs” link is used, it does not mean that it refers to the original resource or even the same resource. These resources are same just form a particular publisher’s point of view. It is another weak point of URI aliases.

To achieve unambiguousness, each resource should have only one URI, defined by the resource owner that takes care of the resource: a person, organization, community, or any other correspondent authority. The Semantic Web ecosystem platform should provide a mechanism for resource creation and correspondent tools that autonomously browse the Web and search for similar resource definitions to initiate a process of ownership detection. The same similarity search engine can be used for the required resource URI search to make the proper reference. Thus, the main information about the resource is provided by the owner (responsible authority). Data can be updated by the owner on request from third parties with correspondent support from the ecosystem platform. At the same time, we cannot deny the possibility for others to have their own opinion and to provide additional information (define new values for the resource properties) about a resource they do not own. However, the owner may ignore this information. In this case, any additional data linked to the resource by a third party should be marked by the ecosystem platform as unverified information and the correspondent owner will be notified about it. All such unverified information should be stored by the ecosystem platform under the correspondent contextual description [25] and be available under these contexts for other users.

Thus, we still allow different views and opinions to be expressed and, at the same time, we avoid multiple URI aliases. We do not need to create a centralized naming authority to assign URIs that would introduce administrative and bureaucratic overhead. Instead of it, the Semantic Web ecosystem will automatically search for similar resources and apply correspondent actions in case of data duplication. Someone might think that this approach can be a case of “central point of failure”, but it is not. Data is still located in different data sources and, if data warehousing techniques (replication, redundancy of data, etc.) would not help and the main data would be lost, then later on it can be regenerated from distributed locations. Otherwise, if we continue to use aliases and apply hundreds of millions of “owl#sameAs” to express identity links, we will have a huge unmanaged mass of claims of different parties rather than facts and related set of additional claims in correspondent contexts.

B. *URI’s dereference*

The Web is intended to be an information space that may be used by humans as well as machines. There are two different strategies to make URIs that identify real-world objects dereferenceable. Both strategies ensure that objects and documents that describe them are not confused. They also ensure that both humans and machines can retrieve appropriate representations. The strategies are called 303 URIs and hash URIs [26]. The hash URIs method has the advantage of reducing the number of necessary HTTP round-trips, which in turn reduces access latency. Then again, the descriptions of all resources sharing the same non-fragment URI part are always returned to the client together. This leads to large amounts of data being unnecessarily transmitted to the client. The 303 URIs method, on the other hand, is very flexible because the redirection target can be configured separately for each resource. There could be one descriptive document for each resource or one large document for all of them.

In our opinion, it is more reasonable to leave resource URI references in a machine-readable form, especially because we are going to utilize data browsers (machines in this case) to present information to the users. Moreover, we should not limit ourselves to just one human readable representation of the resource. We have to consider different representation forms and views to present data depending on different contexts. Thus, a URI points to an RDF description of the resource. This description contains a set of context dependent statements with a property (for example "#visualRepresentation") that refers to correspondent human readable/viewable resource representation. Now, when the Linked Data browser reaches the resource RDF description, it chooses (depending on the context) the appropriate representation form. After that it shows the resource to the user directly or through appropriate visualization services. Some of the techniques relevant to context-aware information visualization and browsing techniques can be found in our previous works [23] [24] [27] [28].

C. Incoming links: symmetry of the properties

In most cases data is linked in one direction. An RDF triple links one resource to another. Usually, describing a certain resource, only relevant (from the point of view of this resource) resources are linked through correspondent properties. It allows us to browse and discover those linked resources. But what about discovering the original resource? While linking another resource to the original one, we have to ensure that if other resources do not have back links to the original one, then at least the ecosystem collects this information under the correspondent context and makes it available for applications and services. It makes our original resources discoverable from the descriptions of other resources additionally created by the ecosystem. Such incoming links enable the user to navigate backwards with Linked Data browsers. In addition to that, they enable crawlers of Linked Data search engines to discover original resources and continue crawling. Naturally, the "owners" of the resources should be notified by the ecosystem platform manager about additional automatically created descriptions. Later, if the "owner" wants, such description can be added to the main resource description with the support of correspondent ecosystem tools.

D. Context-sensitive equivalence of ontologies

Another challenge for Linked Data comes from ontologies, more precisely, from the multitude of them created for different domains. Even if the ontologies are defined for different domains, it might happen that some of the properties have the same meaning and users might describe the data differently following different ontologies. In some cases, such equality of the properties might be context dependent and should be treated with respect to contextual conditions. To manage this heterogeneity, the supportive ecosystem should have a registry that contains a context-sensitive definition [25] of the properties' similarity. The process of similarity detection could be done in a semi-automated way by a similarity search engine with an approval from the responsible expert or/and with a help of weighted feedback from the users. Thus, supportive tools of the ecosystem (browser, search engine, etc.) could automatically request such a registry to get the correspondent list of similar properties and provide a better service.

E. Context –aware Policy-based Data Access Control and Similarity-based Data Search

In order to elaborate the mechanisms for context-aware policy-based resource access and contextually related information retrieving, we require a model. This model should present the influence of the contextual information on data search and retrieving process, on the level of relevance of the links and similarity of the resources. Depending on the context, properties become more or less relevant. This gives us different vectors of weights. In the same way, the context might influence data access, data privacy and security issues. Thus, context-dependent policy-based control seems to be a very promising approach, which is able to keep data links flexible, dynamic and controlled at the same time. This approach should allow us not to program a fixed and hardcoded data access control and search system, but to build it with the ability to change the internal structure on the fly when the context is changed [29] [30].

In our opinion, it would be reasonable to extend traditional explicit semantic links within Linked Data with the implicit ones, for example those, which could be automatically derived by various reasoners. Among those, special attention should be paid to the "semantic similarity" links. Usually, when one queries data, one looks for the resource(s), which are "the same" as the one specified in the query. However, often none of them are found. In many cases it makes sense to find a resource "similar" to the target one. Similarity search was always a big issue within many disciplines and it is especially important for Linked Data. Similarity search should also simplify extensive work aimed at recognizing same resources that have different URIs (see subsection A). Usually we see first that some resources look similar and therefore in practice could be the same ones and then check on the identity of the resources. Therefore explicit similarity links between data entities could be discovered as a result of appropriate similarity search procedures. The challenge here is that some resources being very different in one particular context could be considered similar ones in some other context. As one of the results of the UBIWARE project we developed a prototype of a context-sensitive visual resource browser [27] that we use as a basis for the Linked Data browser and similarity search engine.

V. Conclusions

The Linked Data concept provides us with the possibility to create a complete data space for humans, applications and services. Linked Data is essential to actually interconnect the Semantic Web. The paper presents an agent-based ecosystem with the Linked Data infrastructure as a playground for Semantic Web-driven services and applications. Within this paper we reviewed relevant technologies, showed the benefits and highlighted some challenges of the Linked Data infrastructure with possible ways that might provide the solutions. To facilitate proper utilization of the technology and increase the benefits of it, we need technical support in a form of an ecosystem platform with appropriate tools and services that do all the "dirty work" and keep the whole ecosystem in the proper condition. Middleware has traditionally been used to simplify application development by hiding low-level details and by offering generic services that can be reused and configured by application developers. To build such an ecosystem we base our research and development on the UBIWARE platform. The UBIWARE platform is a tool for proactive interoperability of distributed heterogeneous components. Utilizing Semantic Web and Multi-Agent System approaches, the UBIWARE platform makes the Ecosystem proactive and flexible. The paper describes capabilities and tools of the proposed ecosystem.

ACKNOWLEDGMENT

This research is based on activities of the Industrial Ontologies Group in the latest projects in Agora Center and Department of Mathematical Information Technology (University of Jyväskylä, Finland): UBIWARE, iSCOPE and projects under Cloud Software program in TIVIT SHOK. The authors are very grateful to the members of IOG for their fruitful cooperation within this research topic.

REFERENCES

[1] J. Domingue, D. Fensel, and R. González-Cabero, "SOA4All, Enabling the SOA Revolution on a World Wide Scale", In: Proceedings of the 2nd IEEE International Conference on Semantic Computing (ICSC 2008), August 4-7, 2008, Santa Clara, California, USA, IEEE CS Press, 2008, pp. 530-537.

[2] Semantic Web, 2001. URL: http://www.w3.org/2001/sw/, Accessed June 15th 2011

[3] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American 284(5), 2001, pp. 34-43.

[4] A. Ankolekar, M. Burstein, J.R. Hobbs, O. Lassila, D.L. Martin, D. McDermott, S.A. McIlraith, S. Narayanan, M. Paolucci, T.R. Payne, and K. Sycara, "DAML-S: Web Service Description for the Semantic Web", 2002. URL: http://www-2.cs.cmu.edu/~terryp/Pubs/ ISWC2002-DAMLS.pdf, Accessed June 15th 2011

[5] M. Paolucci, T. Kawamura, T.R. Payne and K. Sycara, "Importing the Semantic Web in UDDI", 2002. URL:http://www-2.cs.cmu.edu/~softagents/papers /Essw.pdf, Accessed June 15th 2011

[6] FIPA, "FIPA Interaction Protocol Library Specification Specification", FIPA00025, 2001. URL: http://www.fipa.org/specs/fipa00025/, Accessed June 15th 2011

[7] F. Curbera, M. Dufler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the Web Services Web: An introduction to SOAP, WSDL and UDDI", Internet computing, 2002, pp. 86-93

[8] J. Clabby, "Web Services Executive Summary", 2002. URL: http://www-106.ibm.com/ developerworks/webservices/library/ws-gotcha/?dwzone= webservices, Accessed June 15th 2011

[9] T. Berners-Lee, "Linked Data - Design Issues". 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html, Accessed June 15th 2011

[10] T. Heath and C. Bizer, "Linked Data: Evolving the Web into a Global Data Space" (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool. 2011

[11] W3C, "SPARQL Protocol and RDF Query Language", 2008. URL: http://www.w3.org/TR/rdf-sparql-query/, Accessed June 15th 2011

[12] W3C, "Resource Description Framework", 2004. URL: http://www.w3.org/RDF/, Accessed June 15th 2011

[13] F.M.T.Brazier, J.O. Kephart, H. Parunak, and M.N. Huhns, "Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda," IEEE Internet Computing, vol. 13, no. 3, May/June 2009, doi:10.1109/MIC.2009.51, pp. 82-87

[14] N. Jennings, "An agent-based approach for building complex software systems". Communications of the ACM 44, 4, 2001, pp. 35-41

[15] N. Jennings, "On agent-based software engineering", Artificial Intelligence 117(2), 2000, pp. 277–296

[16] O. Kaykova, O. Khriyenko, D. Kovtun, A. Naumenko, V. Terziyan, and A. Zharko, "General Adaption Framework: Enabling Interoperability for Industrial Web Resources", In: International Journal on Semantic Web and Information Systems, Idea Group, ISSN: 1552-6283, Vol. 1, No. 3, July-September 2005, pp. 31-63.

[17] JADE agent platform. URL: http://jade.tilab.com/, Accessed June 15th 2011

[18] UBIWARE agent platform. URL: http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE_details.htm, Accessed June 15th 2011

[19] A. Katasonov and V. Terziyan, "SmartResource Platform and Semantic Agent Programming Language (S-APL)", In: P. Petta et al. (Eds.), Proceedings of the 5-th German Conference on Multi-Agent System Technologies (MATES'07), 24-26 September, 2007, Leipzig, Germany, Springer, LNAI 4687 pp. 25-36.

[20] A. S. Rao, "AgentSpeak(L): BDI agents speak out in a logical computable language", In: Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world (MAAMAW '96), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996, pp. 42-55.

[21] Agent Factory Agent Programming Language. URL: http://www.agentfactory.com/index.php/Main_Page, Accessed June 15th 2011

[22] FIPA, "FIPA SL Content Language Specification", 2002. URL: http://www.fipa.org/specs/fipa00008/SC00008I.html, Accessed June 15th 2011

[23] O. Khriyenko, "4I (FOR EYE) Technology: Intelligent Interface for Integrated Information", In: Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS-2007), Funchal, Madeira – Portugal, 12-16 June 2007, pp. 278-281

[24] O. Khriyenko, "Context-sensitive Multidimensional Resource Visualization", In: Proceedings of the 7th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2007), Palma de Mallorca, Spain, 29-31 August 2007, pp 147-153

[25] O. Khriyenko and V. Terziyan, "A Framework for Context-Sensitive Metadata Description", In: International Journal of Metadata, Semantics and Ontologies, Inderscience Publishers, ISSN 1744-2621, 2006, Vol. 1, No. 2, pp. 154-164.

[26] L. Sauermann and R. Cyganiak, "Cool uris for the semantic web - w3c interest group note". http://www.w3.org/TR/cooluris/, 2008, Accessed June 15th 2011

[27] O. Khriyenko, "Context-sensitive Visual Resource Browser", In: Proceedings of the IADIS International Conference on Computer Graphics and Visualization (CGV-2008), Amsterdam, The Netherlands, 24-26 July 2008, pp. 227-232

[28] O. Khriyenko, "4I (FOR EYE) Multimedia: Intelligent semantically enhanced and context-aware multimedia browsing", In: Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP-2007), Barcelona, Spain, 28-31 July 2007, pp. 233-240

[29] A. Naumenko, "SEMANTICS-BASED ACCESS CONTROL: Ontologies and Feasibility Study of Policy Enforcement Function", In: J. Filipe and J. Minguillon (Eds.), Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST-07), March 3-6, 2007, Barcelona, Spain, pp. 150–155

[30] O. Khriyenko, S. Nikitin, and V. Terziyan, "Context-Policy-Configuration: Paradigm of Intelligent Autonomous System Creation", In: Joaquim Filipe and Jose Cordeiro (Eds.), Proceedings of the *12th International Conference on Enterprise Information Systems (ICEIS-2010)*, 8-12 June, 2010, Funchal, Madeira - Portugal, ISBN: 978-989-8425-05-8, pp. 198-205.

Bringing Context to Intentional Services

Salma Najar
Centre de Recherche en Informatique-Université Paris1
90, rue de Tolbiac 75013 Paris - France
Citypassenger
1, avenue de l'atlantique 91976 Courtaboeuf – France
Salma.Najar@malix.univ-paris1.fr

Manuele Kirsch-Pinheiro, Carine Souveyet
Centre de Recherche en Informatique-Université Paris1
90, rue de Tolbiac 75013 Paris - France
Manuele.Kirsch-Pinheiro@univ-paris1.fr,
Carine.Souveyet@univ-paris1.fr

***Abstract*—In service-orientation, the notion of service is used in different views. On the one hand, several approaches have been proposing services that are able to adapt themselves according to the context in which they are used. On the other hand, some researches have been proposing to consider user's goals when proposing business services. We believe that these two views are complementary. A goal is only meaningful when considering the context in which it emerges, and conversely, context description is only meaningful when associated with a user goal. In order to take profit of both views, we propose to extend the OWL-S service description by including on it both the specification of context associated with the service and the goal that characterize it**.

Keywords-OWL-S; SOA; intentional service; context aware service.

I. INTRODUCTION

Service-Oriented Architecture (SOA) is a computing paradigm lying on the notion of service as fundamental element for developing software applications [16]. Its key feature is the notion of services, which stands to independent entities, with well-defined interfaces that can be invoked in a standard way, without requiring the client to have knowledge about how the service actually performs its tasks [5].

SOA can be viewed through multiple lenses, from the IT perspective up to business leaders [27]. The notion of service is used on different abstraction levels. Technically, it refers to a large variety of technologies (Web Services, ESB [21], OSGI [15], etc.). On a business level, services are proposed as a way to respond to high-level user requirements.

On the one hand, we can observe a tendency to context-awareness and adaptation on services. Several authors [10][24][25][26] have been proposing services that are able to adapt themselves according to the context in which they are used. These services are usually called context-aware services [10]. Their importance is growing with the development of pervasive and mobile technologies. Context-aware services focus on service adaptation considering the circumstances in which it is requested. However, considerations such as why context is important and what is its impact to the user's needs remain underestimated.

On the other hand, research has pointed out the importance of considering user's requirements on service orientation. Several works [7][13][16][19] proposed to take into account user's goals when proposing business services. According to these works, a service is supposed to satisfy a given user's intention. However, even when considering high level services, as business services, one should consider variability related to context on service execution. Several authors have been considering the influence of context information on business process [20] [22]. This influence remains whenever such processes are implemented through business services. Such services still have to cope with the context in which they are called.

Therefore, we have two separated views of service orientation. First, we have an extremely technical view, which focuses on technical issues needed to execute and adapt service in highly dynamic environments. In the opposite, we have a high level view, which focuses on user's requirements. The latter considers why a service is needed, without necessarily considering how it is executed, neither in which circumstances it is performed. More than the execution context, this high level view ignores the context in which user's goals emerges, while technical view passes over user's goals behind observed context information.

We believe that these two views are complementary and should not be isolated from each other. Fully potential of service orientation will not be reached if we do not consider both points of view: *goal-based services* and *context-aware services*. For us, a goal is only meaningful when considering it in a given context and a context description is only meaningful when associated with a user goal. However, this goal is not a simple coincidence; it emerges because he is under a given context.

In this paper, we propose a semantic description of services that encompass the description of the goals service can satisfy and the context in which this goal is meaningful.

This paper is organized as follows: Section II presents an overview on related work. Section III introduces the notion of goal and its representation, while Section IV presents the notion of intentional service. In Section V, we discuss the notion of context and its representation. In Section VI, we propose a semantic descriptor for intentional and context-aware services. And finally, we conclude in Section VII.

II. RELATED WORKS

A service can be seen as an independent and easily composed application that can be described, discovered and invoked by other applications and humans. In the last decade, the notion of service has evolved, from simple Web services to semantic Web services [12]. Indeed, we could observe an important tendency for semantically describing services, in order to handle potentially ambiguous service descriptions [12]. Such semantic description is based on richer representation languages, mainly OWL-S [11], which provides a comprehensive specification of a service.

A semantic description is one of the building blocks of context-aware services. *Context-aware services* can be defined as services which description is associated with contextual (notably non-functional) properties. Several authors [24][26] have been proposing context-aware services, whose importance is growing with the development of pervasive and mobile technologies. An illustration of this phenomenon is given by [24], who propose improving service modeling, based on OWL-S, with context information (user information, service information and environment information). Suraci *et al.* [24] focus on adapting service composition to the user's requirements concerning context (device capabilities, user's location, etc.). These authors consider that user should be able to specify contextual requirements corresponding to the service he is looking for (availability, location, etc.), as well as to the context provided by the environment (wireless connection, etc.).

Other authors, such as [26], also advocate for representing context requirements when describing context-aware services. Toninelli *et al.* [26] consider that, in pervasive scenarios, users require context-aware services that are tailored to their needs, current position, execution environments, etc. Therefore, service modeling should be improved, including contextual information. Such a semantic modeling contributes not only to handle problems related to service interoperability, but also to consider different aspects of the environment in which the service is executed.

A different point of view is given by [1][7][13], which highlight the importance of considering user's requirements on service orientation. According to them, a service is supposed to satisfy a given user's intention, formalized as a user goal, which becomes central to service definition.

Among these works, [7] and [19] propose a service oriented architecture based on an intentional perspective. Such architecture proposes the notion of *intentional service*, which represents a service focusing on the intention service allows to satisfy rather than the functionality it performs. Besides, Mirbel e*t al.* [13] propose goal-based service discovery mechanisms. They propose a semantic approach guided by the user's intentions, in which user's requests are expressed using semantic Web technologies

None of these works considers the notion of context, contrary to [1], which proposes a goal-based dynamic service discovery and composition framework that uses context information. Nevertheless, context information is used only for filtering the input of the user's request.

All these works represent two different views of service orientation: (i) one view proposing a context-aware based approach, which focuses on the adaptation of services according to the context information; and (ii) a second view focusing on a goal-based approach, proposing high level services, which focuses on user's goals. The first view focuses on service composition on a highly dynamic environment, without considering why service is needed. The latter considers this question without considering context in which this need emerges.

Questions such as "*why a service is useful in a given context*?" or "*in which circumstances a service need raises*?" remain unexplored. For us, *a goal is only meaningful when considering it in a given context and a context description is only meaningful when associated with a goal*. In order to explore both views, we have first to represent them in a semantic way. Thus, we propose a semantic description of services that encompass notions of context and intention.

III. UNDERSTANDING USER GOALS

Several researches in service engineering [4][7][17] focus on the adoption of goal-based approaches from requirements engineering to identify user's requirements and intentions. This vision is the base for several goal-based approaches [1][23], which propose to take into account user's goals when proposing business services.

The term goal has several different meanings. According to [6], a goal is an "optative" statement expressing a state that is expected to be reached or maintained. The intention represents the goal that we want to achieve without saying how to perform it [7]. Bonino *et al.* [1] defines an intention as a goal to be achieved by performing a process presented as a sequence of intentions and strategies to the target intention. Even if they differ, all these definitions let us consider an intention as a *user's requirement representing the goal that a user wants to be satisfied by a service without saying how to perform it.*

To ensure a powerful intention matching, the intention is formulated according to a template [7][19], represented in Fig. 1. In this template, a goal is expressed by a verb, a target and a set of optional parameters, which play specific roles with respect to the verb.

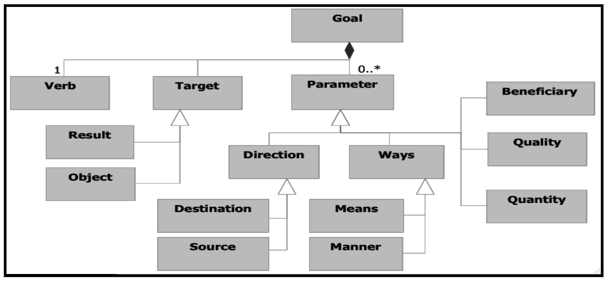


Fig 1. Goal template based on [7]

IV. INTENTIONAL SERVICE AT THE GLANCE

Recently, several authors have considered a direct participation of end user on service specification. Brnsted *et al.* [2] illustrate this tendency by observing several approaches allowing end users to actively interact with service composition specification. However, these authors do not consider whether terminology used by these tools correspond to the user's current vocabulary. The question that emerges here is the following: are these users technical people, who are familiarized with service-oriented technology, or are these users business actors who are totally unaware of technical considerations?

To bridge the gap between high-level business services, comprehensible by the business actors, and low-level software services, understandable by technical people, an intentional description is proposed [7] [19]. This user-centric perspective forms the so-called *Intentional Service Oriented Architecture (ISOA)*. ISOA represents services at a high level of abstraction, referring to the intention they can fulfil rather than the function they perform. Such services, named *intentional services*, are expressed in terms of intentions and strategies to achieve them.

A. Defining intentional services

An intentional service is a service captured at a high-level, in business comprehensible terms and described in an intentional perspective. The intentional service model (ISM) [7] [19] associates to each service an intention it can satisfy. It is composed of 4 facets, represented in Fig. 2, namely the *service interface*, the *service behaviour*, the *service composition* and the *QoS*.

The *service interface* represents the service that permits the fulfilment of an intention, given an initial situation and terminating in a final situation. The *service behaviour* specifies the pre and post conditions that represent the sets of initial states required by the service for the goal achievement, and the set of final states resulting from goal achievement. The *service composition* represents the possibility of composing more complex goals by combining lower abstraction level goals. Next section gives more precisions about service composition. Finally, the *QoS* introduces the non-functional dimension of service. It represents the quality requirements associated with intentional services.

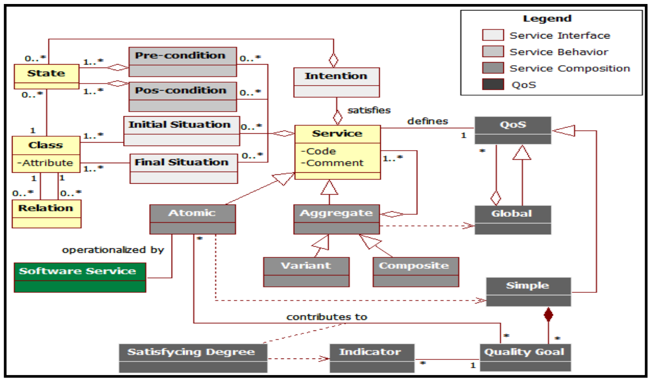


Fig 2. Intentional Service Model (ISM) [19]

B. Composing intentional services

The intentional service model emphasises variability on the satisfaction of its corresponding intention. It allows the variability through the service composition. In the ISM model, an intentional service can be *aggregate* or *atomic*. Aggregate services represent high-level intentions that can be decomposed in lower level one, helping business people to better express their strategic/tactical intentions.

Intentional composition admits two kind of aggregate services: a *composite* and a *variant*. While composite services reflect the precedence or succession relationship between two intentions, variant service correspond to the different manner to achieve an intention. This need for variability is justified by the need to introduce flexibility in intention achievement.

According to [19], atomic services are related to operationalized intentions and can be fulfilled by SOA functional services. Atomic intentional services are then operationalized by software services. In contrast, aggregate services have high-level intentions that need to be decomposed in lower level ones till atomic intentional services are found.

Nevertheless, this vision does not consider the evolution of service technology, which can stand now for small pieces of software encapsulating reusable functionalities, as well as for large legacy systems, whose complex process are hidden by technologies such as Web Services or ESB [21]. By considering that only atomic services can be operationalized by software service, ISOA architecture limits the reuse of such legacy systems under an intentional approach.

In this paper, we consider that both atomic and aggregate intentional services can be operationalized by software service, which can be also atomic or composite. As a consequence, both technical and intentional compositions are possible independently, allowing more powerful constructions. Section VI describes how both can coexist in the proposed service semantic description.

V. Describing Context Information

In the last decade, an important change has been performed on the way we work and on the way technology support us. We pass from a quite static model, in which people use to interact with business process only during their "work time" in well-defined circumstances (in their offices, with their desktop computers) to "mobile worker" model. With the evolution of mobile technologies, and notably smartphones, this static model does not fit anymore.

As a consequence of this evolution, information systems should now consider not only the tasks a user can (or must) perform, but also the context in which such user finds himself when performing an action. Context information corresponds to a very wide notion. It is usually defined as any information that can be used to characterize the situation of an entity (a person, place, or object considered as relevant to the interaction between a user and an application) [3]. The notion of context is central to context-aware services that use it for adaptation purposes. Context information can stand for a plethora of information, from user's location, device resources [18], up to user's agenda and other high level information [8]. Nevertheless, in order to perform such adaptation processes, context should be modelled appropriately. The way context information is used depends on what it is observed and how it is represented. The context-adaptation capabilities depend on the context model [14].

Different kinds of formalism for context representation have been proposed. Nevertheless, an important tendency can be observe on most recent works: the use of ontology for context modelling [14]. According to [14], different reasons motivate the use of ontologies, among them their capability of enabling knowledge sharing in a non-ambiguous manner and its reasoning possibilities. This tendency follows the evolution of context-aware services, which adhere, in their majority, to a semantic description of such services. In this paper, we also adhere to this tendency, adopting an ontology-based context modelling based on [18].

VI. Putting Everything Together: Describing Context-Aware Intentional Services

The latest research in service oriented computing recommends the use of the OWL-S for semantically describe services [24]. Even if OWL-S is tailored for Web services, it is rich and general enough to describe any service [24]. OWL-S [11] defines web service capabilities in three parts representing interrelated sub-ontologies named service profile, process model and grounding. The *service profile* expresses what the service does. It gives a high-level description of a service, for purposes of advertising, constructing service requests and matchmaking. The *process model* answers to the question: how is it used? It represents the service's behaviours as a process and describes how it works. Finally, the *grounding* maps the constructs of the process model onto detailed specification of message formats, protocols and so forth (often WSDL).

OWL-S represents a flexible and extensible language, as demonstrated by works such as [9][24]. Similar to these works, we propose to extend service description in OWL-S by including information concerning both context and goal that characterize a service.

A. Describing service intentions in OWL-S

According to an intentional perspective, a user requires a service because he has a goal that the service is supposed to satisfy. Hence, the importance of considering user's goals emerges on service orientation. Such goal is formalized as an intention, which becomes central to service definition.

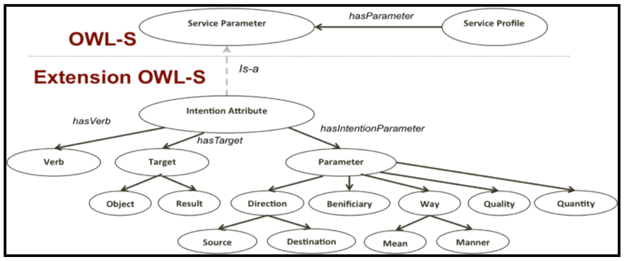
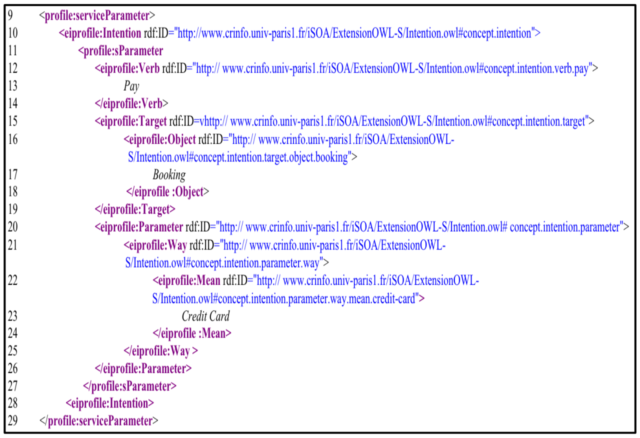


Fig 3. Service intention Description in OWL-S profile

OWL-S profile provides features to expose characteristics of a service through service parameter [11]. We propose to extend service profile by adding a new parameter named *intention* attribute, which describes the intention associated with the service (see Fig. 3). This parameter is described using the template *"Verb, Target, Parameters"* (see Section III). Thus, the intention attribute of a service assures to the service an intentional interpretation.

Fig. 3 details elements we added to OWL-S profile description. First of these elements is the *intention* attribute itself, which is an extension from OWL-S service parameter. A service intention has three properties: a *verb*, a *target* and a *parameter*. A *verb* characterizes the user's intention. Possible verbs can be organized in a verb ontology that recognizes significant verbs for a given community. A *target* indicates either a *result* from the satisfaction of the goal, or an *object* that exists before the achievement of the goal. Finally, a *parameter* represents additional information needed by the *verb*.



```
9       <profile:serviceParameter>
10          <eiprofile:Intention rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-S/Intention.owl#concept.intention">
11              <profile:sParameter
12                  <eiprofile:Verb rdf:ID="http:// www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-S/Intention.owl#concept.intention.verb.pay">
13                      Pay
14                  </eiprofile:Verb>
15                  <eiprofile:Target rdf:ID=vhttp:// www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-S/Intention.owl#concept.intention.target">
16                      <eiprofile:Object rdf:ID="http:// www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                         S/Intention.owl#concept.intention.target.object.booking">
17                          Booking
18                      </eiprofile :Object>
19                  </eiprofile:Target>
20                  <eiprofile:Parameter rdf:ID="http:// www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-S/Intention.owl# concept.intention.parameter">
21                      <eiprofile:Way rdf:ID="http:// www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                        S/Intention.owl#concept.intention.parameter.way">
22                          <eiprofile:Mean rdf:ID="http:// www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                            S/Intention.owl#concept.intention.parameter.way.mean.credit-card">
23                              Credit Card
24                          </eiprofile :Mean>
25                      </eiprofile:Way >
26                  </eiprofile:Parameter>
27              </profile:sParameter>
28          <eiprofile:Intention>
29      </profile:serviceParameter>
```

Fig 4. Example of describing service intention in OWL-S

Fig. 4 illustrates this extension through an example of service profile that includes the intention attribute. This example presents a *booking service,* which satisfies the intention *booking payment by credit card* (lines 10-28). The lines 12-14 describe the verb *pay*, which describes the intention. The target of the intention is represented by the object *booking,* described in the lines 15-19. This intention has, as a *parameter,* the *mean* of the intention represented by the *credit card* in this example (lines 22-24).

B. Describing contextual information in OWL-S

A goal that a user wants to satisfy is not a coincidence; it emerges because the user is in a given location or under a given context. In our opinion, a goal is only meaningful when considering it in a given context and a context description is only meaningful when associated with a goal. According to this, we propose to extend the service profile to allow service provider to define context information that characterize an intentional service.

For instance, let us consider a *parachute jump booking service* that enables users to browse, search for, and reserve a parachute jump in different situations and according to the user preference. This service can be particularly designed considering client devices with high screen resolution and flash technologies to show video and tutorials about the parachuting; a second implementation of the same service can be designed considering, for example, a particular user profile (e.g., adult users). Such contextual information can be considered as part of the service description, since it indicates situations to which the service is better suited. According to [9], context information cannot be statically stored on the service profile due to its dynamic nature. Context properties related to service execution can evolve (e.g., server load may affect properties of services running on it), whereas service profile is supposed to be a static description of the service.

Thus, in order to handle dynamic context information on static service description, we adopt the approach [8] to enrich OWL-S service profile with a *context* attribute, which represents a URL pointing to context description file. Since context information is dynamic and cannot be statically stored on the service profile description, we opt to describe context element in external file to allow service provider to easily update such context information related to the service description itself. The context description of a service describes, from the one side, the situation status of the requested service (environment in which the service is executed), and from the other side, the contextual condition (requirement) to execute the service.

C. Composing intentions

Intention and context attributes described above intent to expose both aspects of a service notably for discovery purpose. Thanks to the OWL-S extension we propose, a service can be discovered either by intention it can satisfy, or by the context associated with this intention. In addition to these aspects, a third aspect should be exposed: the service variability. Such variability is expressed, in the intentional perspective, by the composition of intentional services indicating the decomposition of the service goal on lower level goals. Thus, while technical composition of a service, described in OWL-S process model, represents software components that are combined to supply service operations, intentional composition represents not only lower level goals necessary to satisfy service goal, but also different possibilities for satisfying this goal. Technical composition supplies technical elements necessary for service execution, while intentional composition provides an understanding, from final user's point of view, of the service and the diverse forms of satisfying service goal. Thus, we propose to extend OWL-S process model by including the specification of an intentional service process (see Fig. 5).

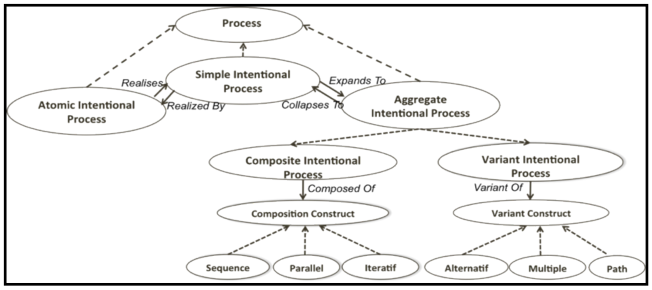


Fig 5. Composing intentions in OWL-S process Model

Fig. 5 presents the extension we propose for the process model. This extension considers two kinds of process: the atomic intentional process and the aggregate intentional process. It considers also a simple intentional process, which is used to provide an abstracted view that can be atomic or aggregate. A simple intentional process is realized by an atomic intentional process and expands into an aggregate intentional process. An aggregate intentional process can be either a composite intentional process or a variant intentional process.

The composite intentional processes reflect the precedence/succession relationship between their intentions. Such relationships are specified using composition constructs such as *Sequence*, *Parallel* and *Iterative*. The composition represents a *sequence* in which there is a sequential order between component processes, or a *parallel* in which components can run in parallel. The *iterative* construct is used when the satisfaction of a goal may require iterative execution of a given set of actions.

The variability is represented by the variant intentional process, which uses constructs such as *multiple*, *alternative* and *path*. The *multiple* construct offers a non-exclusive choice in the realization of the goal. It groups multiple simple processes, among them, at least one will be chosen. The *alternative* construct represents a process with an alternative choice that regroups several simple processes that are mutually exclusive. It builds a new process of the same level of abstraction but of higher granularity. And finally, the *path* construct offers a choice in how to achieve the goal of the aggregate process by offering composite processes that are mutually exclusive.

```
<eiprocess:CompositeIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
S/services/MakeParachuteJumpBooking">
    <eiprocess: CompositeIntenionalProcessID> MakeParachuteJumpBooking </eiprocess: CompositeIntenionalProcessID>
    <eiprocess: CompositeIntenionalProcessName> Make a parachute jump booking </eiprocess: CompositeIntenionalProcessName>
    <eiprocess: ComposedOf>
            <eiprocess: Sequence>
                        <process: Components rdf:parseType="Collection">
                        <eiprocess: VariableIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                        S/services/PayParachuteJumpBooking">
                        <eiprocess: AtomicIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                        S/services/ReserveDateParachuteJumpBooking">
                        <eiprocess: AtomicIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                        S/services/TakeInsurance">
                        </process: Components>
            </eiprocess: Sequence >
    </eiprocess: ComposedOf>
</eiprocess: CompositeIntenionalProcess >
```

Fig 6. Example of OWL-S Intentional composition: Composite Intentional Process

For instance, let us consider the example a service named $S_{\textit{Confirm parachute jump booking}}$, which intents making a confirmed parachute jump booking. It is described as a variant service that represents a *path* between the composite service $S_{\textit{Get a rewarded parachute jump booking}}$ and the service $S_{\textit{Make parachute jump booking}}$ (see Fig. 6). This latest is composed of a sequence of the variant service $S_{\textit{Pay parachute jump booking,}}$ the atomic services $S_{\textit{Reserve the date of the parachute jump}}$ and service $S_{\textit{take an insurance}}$ (see Fig. 7).

```
<eiprocess: VariableIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
S/services/confirmParachuteJumpBooking">
    <eiprocess: VariableIntenionalProcessID> ConfirmParachuteJumpBooking   </eiprocess: VariableIntenionalProcessID>
    <eiprocess: VariableIntenionalProcess Name> Confirmation parachute jump booking </eiprocess:
VariableIntenionalProcessName>
    <eiprocess: VariantOf>
            <eiprocess:Path>
                    <process:Components rdf:parseType="Collection">
                            <eiprocess: CompositeIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                            S/services/getRewardedParachuteJumpBooking">
                            <eiprocess: VariableIntenionalProcess rdf:ID="http://www.crinfo.univ-paris1.fr/iSOA/ExtensionOWL-
                            S/services/MakeParachuteJumpBooking">
                    </process: Components>
            </eiprocess: Path>
    </eiprocess:VariantOf>
</eiprocess:VariableIntenionalProcess >
```

Fig 7. Example of OWL-S Intentional composition: Variant Intentional Process

Thanks to the OWL-S extension proposed here, we enable the description of intentional composition, from final user's point of view. This extension exposes the variability representing different manners to satisfy user's goals. The intentional composition description allows a service discovery guided by intention, presented at a high level.

VII. Conclusions and Future Works

In this paper, we considered context-aware and goal-based service orientation as complementary approaches that should not be isolated from each other. We explain our belief that a goal is only meaningful when considering it in a given context and a context description is only meaningful when associated with other goal. We propose, consequently, to enrich OWL-S service description, by including the description of the goals service can satisfy as well as context in which this goal is meaningful, context in which service is (or can be) executed. From the one side, we propose to enrich service description with knowledge about goals and composition of goals that are meaningful for final users, who request the service. From the other side, we propose to enrich this service description with context information necessary for adapting such service. By proposing such a semantic description of service, we enable the expression of services that can adapt themselves to context of use and that represent a formulated user's requirements. By exposing both aspects of a service, we develop a context-aware goal-based service oriented framework. This framework is currently under evaluation.

References

[1] L.O. Bonino da Silva Santos, G. Guizzardi, L.F. Pires, and M. Van Sinderen, "From User Goals to Service Discovery and Composition," ER Workshops, pp. 265-274, 2009.

[2] J. Brnsted, K. Hansen, and M. Ingstrup, "Service Composition Issues in Pervasive Computing," IEEE Pervasive Computing, vol. 9 n° 1, pp. 62 -70, 2010.

[3] A. Dey, "Understanding and using context, Personal and Ubiquitous Computing," vol. 5 n°1, pp. 4-7, 2001.

[4] J. Gomez, M. Rico, and F. Garcia-Sanchez, "GODO : Goal Oriented Discovery for Semantic Web Services," WIW Workshop on WSMO Implementations, 2004.

[5] V. Issarny, M. Caporuscio, and N. Georgantas, "A Perspective on the Future of Middleware-based Software Engineering," In: Briand, L. and Wolf, A. (Eds.), Future of Software Engineering 2007 (FOSE), ICSE (Conf on Software Engineering), IEEE-CS Press.

[6] M. Jackson, "Software Requirements and Specifications: A lexicon of practice, principles and prejudices," Addison Wesley Press, 256, 1995.

[7] R.S. Kaabi and C. Souveyet, "Capturing intentional services with business process maps," RCIS, pp. 309-318, 2007.

[8] M. Kirsch-Pinheiro, J. Gensel, and H. Martin, "Representing Context for an Adaptative Awareness Mechanism," G.-J. de Vreede; L.A. Guerrero, G.M.Raventos (Eds.), LNCS 3198 - X Workshop on Groupware (CRIWG 2004), pp. 339-348, 2004.

[9] M. Kirsch-Pinheiro, Y. Vanrompay, and Y. Berbers, "Context-aware service selection using graph matching," 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), ECOWS. CEUR Workshop proceedings, vol. 411, 2008.

[10] Z. Maamar, D. Benslimane, and N.C. Narendra, "What can context do for web services?," Communication of the ACM, vol. 49 n° 12, pp. 98-103, 2006.

[11] D. Martin, M. Paolucci, S. Mcilraith, M. Burstein, D. Mcdermott, D. Mcguinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. Sycara, "Bringing Semantics to Web Services: The OWL-S Approach," Cardoso, J. & Sheth, A. (Eds.), SWSWPC 2004, LNCS 3387, Springer, pp. 26-42, 2004.

[12] S.A. Mcilraith, T.C. Son, and H. Zeng, "Semantic Web Services, IEEE Intelligent Systems," vol. 16, pp. 46-53, 2001.

[13] I. Mirbel and P. Crescenzo, "From end-user's requirements to Web services retrieval: a semantic and intention-driven approach," J.-H. Morin, J. Ralyte, M. Snene, "Exploring service science", First International Conference, IESS 2010, LNBIP 53, Springer, pp. 30-44, 2010.

[14] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet, and S. Nurcan, "Semantic representation of context models: a framework for analyzing and understanding," J. M. Gomez-Perez, P. Haase, M. Tilly, and P. Warren (Eds),1st Workshop on Context, information and ontologies (CIAO 09), European Semantic Web Conference (ESWC), ACM, pp. 1-10, 2009.

[15] OSGi Alliance, http://www.osgi.org/ (Jan 2011).

[16] M.P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: A Research Roadmap," Int. J. Cooperative Inf. Syst. vol 17 n° 2, pp. 223-255, 2008.

[17] L. Penserini, A. Perini, and A. Susi, "High Variability Design for Software Agents: Extending Tropos," ACM Transaction on autonomous and Adaptative Systems, vol.2 n°4, 2007.

[18] R. Reichle, M. Wagner, M. Khan, K. Geihs, L. Lorenzo, M. Valla, C. Fra, N. Paspallis, and G.A. Papadopoulos, "A Comprehensive Context Modeling Framework for Pervasive Computing Systems," In 8th IFIP Conf on Distributed Applications and Interoperable Systems (DAIS), Springer.

[19] C. Rolland, M. Kirsch-Pinheiro, C. Souveyet, "An Intentional Approach to Service Engineering," IEEE Transactions on Service Computing, vol.3 n°4, pp. 292-305, 2010.

[20] M. Rosemann, J. Recker, and C. Flender, "Contextualization of Business Processes," Int. J. Business Process Integration and Management, vol. 1 n°1/2/3, 2007.

[21] W. Roshen, "SOA-Based enterprise integration: a step-by-step guide to services-based application integration," McGraw Hill, 2009.

[22] O. Saidani and S. Nurcan, "Towards Context Aware Business Process Modeling," 8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), CAiSE'07, 2007.

[23] M. Stollberg and B. Norton, "A Refined Goal Model for Semantic Web Services," IEEE, Second International Conference on Internet and Web Applications and Services (ICIW'07), pp. 17, 2007.

[24] V. Suraci, S. Mignanti, and A. Aiuto, "Context-aware Semantic Service Discovery," 16th IST Mobile and Wireless Communications Summit, pp. 1-5, 2007.

[25] N. Taylor, P. Robertson, B. Farshchian, K. Doolin, I. Roussaki, L. Marshall, R. Mullins, S. Druesedow, and K. Dolinar, "Pervasive Computing in Daidalos, Pervasive Computing," vol. 10 n° 1, pp. 74 - 81, 2011.

[26] A. Toninelli, A. Corradi, and R. Montanari, "Semantic-based discovery to support mobile context-aware service access," Computer Communications, vol.31 n° 5, pp. 935-949, 2008.

[27] R. Welke, R. Hirschheim, and A. Schwarz, "Service-oriented architecture maturity," IEEE Computer, vol. 44 n° 2, pp. 61-67, 2011.

Web Services with Java EE 6: An Example Using Planning in Reverse

Keith Ballantyne
University of Maryland Baltimore County, USA
kb12@umbc.edu

Abstract—**This paper describes a Java-centric approach to the development of a service-oriented architecture. It introduces many of the technologies readily available within Java to realize a Web service architecture, and explains some of the difficulty encountered during the realization of specific Web services necessary to implement a strategic planning process. Preliminary results indicate that robust support of service-oriented architecture is available using Java technologies, but efficiently meeting non-functional requirements, such as security and platform-independence, pose design challenges for the developer.**

Keywords-**Service-Oriented Architecture; J2EE; Web Services; Information as a Service; Planning In Reverse.**

I. Introduction

In the early part of 2011 this paper's author was approached by faculty members from Alvernia University for help in the construction of an application that would assist the users of a process they had developed. Concurrently, the author had enrolled in a graduate level service-oriented architecture (SOA) course intended to expose students to the utility of constructing applications based on services. The nature of the process defined by the Alvernia faculty lent itself well to a service-oriented approach. Ballantyne, et al. [1] outlines the need for a tool that actively incorporates all employees into the planning process.

> This book was written so that organizations – large and small, private and public, for profit and not for profit school systems or colleges and universities – can implement a process that includes all stakeholders and employees in a meaningful role in planning. It is designed so that all can be engaged in the process and become part of the organization's long-term viability by providing short-term observations.

After reading the book and conferring with its authors, several key features of the process became evident. Successful implementation relies on gathering information that is distributed among many stakeholders, resulting in a need for different types of client-side interactions. Both small and large organizations must be supported with equal utility, making scalability a key non-functional requirement. As an organization grows more adept at applying the process to its unique needs, the tool must change to accommodate the new knowledge. Finally, interaction must be ubiquitous, requiring easy access from a diverse collection of client hardware. Individually, each feature can be realized without giving much consideration to the architecture, but when taken as a whole, a service-oriented approach provides clear benefit. Adopting an *Information as a Service* approach [2] to maintaining data, and ensuring *loose coupling* between the client and business logic [3] provide both scalability and ubiquity in the final solution. A service-oriented approach also provides a path for future growth as the process matures within the organization, both through refinement and *service composition* [4].

The balance of this paper details the construction of the resulting system and some of the challenges faced along the way. It is organized as follows: Section 2 presents an overview of the Planning In Reverse as outlined by its authors, Section 3 provides an overview of Java Enterprise Edition 6 Web Services enabling technologies, Section 4 describes the current progress on implementation and pitfalls encountered, and finally conclusions and areas for future work are presented in Section 5.

II. Planning In Reverse

Scott Ballantyne, brother of this paper's author, along with Beth Berrett and Mary Ellen Wells published a book in 2011 titled *Planning in Reverse: A viable approach to organizational leadership* [1]. Citing many examples of failed strategic planning processes within organizations, the text outlines an alternative approach to strategic planning aimed at continuously updating an organization's strategic plan based on observation of current events. The process outlined in the book begins with an employee recording an observation of an event, and guides strategic planners in assessing the event, developing a plan to respond to the event, and implementing the developed plan. Planning In Reverse's authors were interested in developing software that could support the collection of information and assist strategic planners in keeping track of the process flow.

A. Process Description

At its core, PIR is a tool to facilitate the adaptation of long-term strategic plans to internal and external events that may occur. It is a methodology based on active participation of individual employees in the strategic planning process. The methodology described in the text includes a process of evaluation, implementation, and integration of the resulting strategy into the operational environment of an organization.

The process consists of several basic process steps.

- The initial step starts when an employee observes something they deem significant. News stories, newly available technologies, changes in world affairs, and even casual conversations that an employee feels could potentially impacting the organization are recorded as an *implication scan*.
- Once recorded, these *implication scans* are evaluated by a committee of strategic planners to determine their impact. Impact can be significant or insignificant, and may also be categorized as internal or external. Implications that are deemed to have a material impact are advanced to the next stage for further action.
- *Itemized action plans* are developed to address the implications. These itemized action plans define a sequence of steps for the organization to undertake to actively manage the impact. The plans include both steps to take and the logistics necessary to ensure the plan is successful.
- Finally, the plan is rolled out and integrated into the organization.

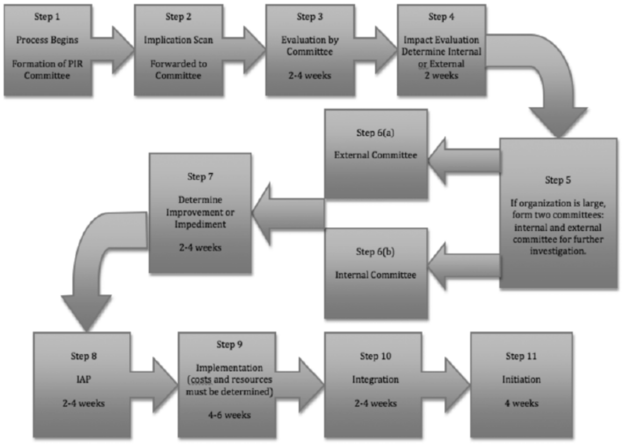


Figure 16.1. Timeline for implementation. PIR = planning in reverse.

Fig. 1. Planning in Reverse, [1]

.

Figure 1 shows a more detailed view of the process outlined within the text. Depending on the size organization, separate committees may be formed to better meet demand. Also note that implication scans can result in both positive (improvements) and negative (impediments) that must be dealt with.

B. PIR Example

To better understand the process flow consider the following example. An employee reads a news article that oil prices are likely to rise over $100 a barrel for the next 12 months. The employee submits an implication scan. The scan is entered into the system, and the next time the impact evaluation committee meets, they review it. They determine that higher energy prices will increase the overhead rates for their factory. These increased overhead rates will make the product they produce more expensive. The additional expense means their product will not be price competitive within the market. Realizing that they are not going to meet their annual operating profit, they develop an itemized action plan to increase the sales of an alternate product that will remain price competitive while reducing the energy usage throughout the factory. This itemized action plan is subsequently developed into an implementation plan by considering the logistics needs and defining the timeline. Finally, the implementation plan is integrated into the organization, and they are ultimately able to meet their annual operating targets.

III. Web Services

The term *web services* broadly refers to service-oriented architectures where services are provided over transport mechanisms such as HTTP, and service clients are available as webpages. As an architectural concept, SOA provides *loose coupling* [5] between the implementation of business logic and the consumers of those implementations. This loose coupling also allows significant capacity for scaling, either through the distribution of service endpoints across multiple platforms, or through the scaling of business logic implementation using multiple cores, distributed computation, or cloud computing. Prolific use of Web services is largely the result of standards that define the mechanisms of interface definition and service invocation.

A. Web Services Definition Language

Web Services Definition Language (WSDL) [6] is the primary mechanism used to define the interface to Web services. Written using the Extensible Markup Language (XML), WSDLs provide a complete encapsulation of the service interface available to the client. WSDL XML documents contain a complete description, including the methods available from the service and the data types that are exchanged. The key elements within a WSDL are defined below. The collection of information is a complete interface specification, allowing programmatic discovery and use of the service.

PortType defines operations and their associated **Messages**.

Types references an XML schema document used to define datatype elements and the namespace.

Message defines a Web service method and **Parts** of the message.

Parts identify parameters used in invoking the service method.

Binding binds each **Operation** with its transport mechanism and datatypes.

Service binds the service name to the port and the physical location at which the service may be found.

B. Simple Object Access Protocol

Simple Object Access Protocol[1] [7] (SOAP) is an XML specification that defines the structure of information passed to and from Web service *endpoints*. Specified by the W3C organization, the protocol is both prolific and well understood.

[1] More recent versions of the specification have dropped the phrase, referring only to the acryonym SOAP

SOAP messages provide client-initiated bidirectional communication with a service. A request is sent from the client to the Web service endpoint, and a SOAP response containing the information is returned from the Web service endpoint to the client. Whereas WSDLs specify the interface to the service, SOAP is used to transmit and receive service data.

C. Java support of Web Services

Traditionally, Web services have been developed using a manual process. WSDLs were constructed by hand, requiring a manual specification of each service endpoint and each operation. During development, the WSDL was manually updated each time a change to an operation was made. Since the resulting WSDL file is machine readable, a logical progression was the development of tooling that generated these files automatically and contained a mechanism to ease deployment. Java provides a robust implementation of all of the necessary components for Web service development.

D. Java 2 Enterprise Edition 6

The Java 2 Enterprise Edition 6 (J2EE 6) [8] specification for enterprise-level Java ensures that the requisite facilities for Web service development and deployment exist within the platform. A comprehensive tutorial is available from the Oracle website [9]. This tutorial includes information on nearly every enterprise edition feature, including Web service development. Enterprise edition 6 is more than a simple collection of compiler tools. This standard specifies services, libraries, and many other facilities that are useful for enterprise-level service deployment. J2EE 6 is not a specific implementation, but rather a specification that certified implementations comply with. Until recently, compliant implementations were only available commercially. However, with the release of GlassFish Version 3 [10], an open source J2EE 6 compliant implementation became available.

Java enterprise edition Web services are defined in JSR-109 [11]. This specification includes detailed information about developing and consuming Web services within Java code. The specification details how WSDLs and the services themselves are accessed. A related specification, JSR-172 [12], details the same thing for J2ME devices.

1) GlassFish: The GlassFish server bundles all of the necessary elements to deploy Java enterprise applications, including Java Web services applications. GlassFish provides administration utilities, application containers, extensive scalability, database management and connection facilities, and centralized user management. As a framework for deploying large-scale enterprise applications, GlassFish contains all of the necessary components in a single installation package. The server itself may be deployed on both Windows and Linux platforms, and is agnostic to the specific version or architecture of the host operating system.

E. Java Architecture for XML Binding

Consumption of Web services in Java relies on the Java Architecture for XML Binding (JAXB) client programming model [13]. This model provides a realization for transmission and reception of both Java Web service data and Java remote procedure call invocation. Though Web services under Java can be produced and consumed using Java remote procedure calls, the implementation within PIR relies solely on the JAXB Web services (JAX-WS) implementation.

JAXB exposes several core functions that are necessary in any Web services realization using Java. First, it provides annotation-driven binding that directly maps XML elements into Java objects. This binding is extensible, allowing implementers to alter both marshaling and un-marshaling of data via callback methods. Second, JAXB provides intrinsic Java type-to-XML mapping, combined with facilities to extend the mapping as necessary. Finally, JAXB provides facilities that allow customization of the XML schema to Java representation.

1) JAX-WS: As noted above, PIR relies solely on the JAX-WS implementation specified in JSR-224 [14]. This implementation includes both client and service facilities to produce and consume Web services. Typical usage involves annotating code using a special syntax to define the Web service, operations within the Web service, and the datatypes used in connection with the web service. A code snippet showing a small portion of the annotation is included below.

```
@WebMethod(operationName = "getLateImplications",
       action="getLateImplications")
public List<Implications> getLateImplications(
       @WebParam(name = "userID") int userID,
       @WebParam(name = "daysLate") int daysLate) {
```

Note the **@WebMethod** and **@WebParam** annotations above. These provide all of the information necessary for the JAX-WS architecture to properly generate the operations associated with the Web service. The annotations significantly reduce the amount of effort required to define the WSDL and SOAP interfaces for the service. Instead, the developer is free to focus on implementing the appropriate logic required within the service. At compile time, the annotations are parsed, and proper WSDL, XSD, and SOAP envelopes are automatically constructed.

Consuming a Web service is even easier. In either the Eclipse [15] or NetBeans [16] integrated development environments, the Web service can be selected from the tree hierarchy on the left pane and simply dropped into a Java class file. In both cases the result of the operation is an automatically generated stub that the developer can call directly. Both integrated development environments are sophisticated enough that a change to the service (WSDL) will be automatically reflected when the IDE is updated. The developer need only delete the prior stub and repeat the drag-and-drop operation. Unlike earlier manual Web service development, modern tools provide both simplicity and expediency when constructing service-oriented architectures.

F. Java Persistance Application Programming Interface

Among its many features, the J2EE environment provides several methods of connecting to a database. GlassFish is equipped with Java Database Connectivity (JDBC) connectors to many of the most popular databases. Even when the connector is not present in the base installation (such as the MySQL Connector/J [17]), installation is simply a matter of copying a Java archive file (JAR) into the appropriate directory. Once a JDBC driver is installed, the GlassFish server provides several mechanisms of allowing applications within it to access the database.

Though any application within GlassFish is capable of direct connection to a database, the most popular mechanism is the use of a connection pool. Connection pools allow any application that shares the same data source to use a commonly specified connection. An application requiring access to the data is pointed at the connection pool rather than a specific JDBC connection. In this way, server administrators can update the entire installation to a new database by simply updating the server's connection pool settings. No change is required to the service code.

JDBC connections allow direct queries of relational databases. Though this is frequently sufficient for data manipulation, it results in stronger coupling between the underlying database query engine and the application. J2EE includes other mechanisms to persist data within the database. The Java persistence Application Programming Interface (JPA) [18] is used throughout the PIR service implementation when possible. In addition to providing the functions typically necessary when manipulating a relational database, JPA represents the database rows as Java objects (called *entities*). Both Eclipse and NetBeans allow automatic construction of these entity objects by directly querying the database. The automatically generated models include many-to-one and one-to-many object relationships. Hence the object representation within Java is identical to the relationship within the database. This enables a cohesive, object-centric approach to data management within a Java enterprise application.

G. Security

One of the most difficult aspects of Web service development for PIR relates to securing the Web service. In 2004 the Web Services Security (WS-Security) specification was produced by OASIS [19]. Subsequently updated in 2006, the document details how SOAP messages may be secured. Most Web services are still secured by using the security mechanism built into the underlying web server. *HTTP basic auth* is frequently used as the identity mechanism. With or without secure socket layer (SSL) support, this basic mechanism presents several problems for the realization of PIR. Most notably, authentication to any service operation is cumbersome to an operator. Without some form of persistence, each SOAP request would require reentry of the username and password. JAXB provides mechanisms that allow the client to specify both username and password, but it still leaves several security weaknesses in Web service deployment. The WS-Security specification addresses these issues by standardizing several newer mechanisms of security within Web services.

Within the PIR application it is important that any call to the database via a Web service is secure. This means that the data transmitted and received via the Internet must be secure from a man-in-the-middle attack. The WS-Security specification enables both X.509 certificates and Kerberos tickets to be used for this purpose. When using security mechanisms such as these, the entire body of the SOAP message is encrypted using the ticket then decrypted by the client. Thus the certificate mechanism provides acceptable security for the protection of corporate strategic planning data.

IV. Implementation

In order for the PIR process to be successfully implemented, two critical objectives must be met. First, it must be easy for an employee to submit an implication scan. The PIR authors desire a ubiquitous application, with broad availability to every member of an organization. The intent is that an employee, regardless of their location, can readily submit a scan into the system. The second objective is that both the employee and the organization's committees are informed of the information and given frequent feedback about progress. PIR requires accountability for ensuring that every implication scan is processed in a timely manner. The process authors want every participant to be aware that their contribution is being considered and acted upon, and that submitters know the current status of their implication scans. These requirements, along with others from alternate stakeholder perspectives, are enumerated below.

A. Threshold Requirements

THR(1) The service(s) shall provide individual contributors with rapid and ubiquitous capability to submit implication scans.

THR(2) The service(s) shall provide individual contributors with routine updates on the progress of their implication scans throughout the process.

THR(3) The service(s) shall provide impact evaluators with the facilities necessary to manage implication scan evaluation.

THR(4) The service(s) shall provide impact evaluators deadline information for implication scan evaluation per process temporal constraints.

THR(5) The service(s) shall provide the capability to develop itemized action plans (IAP).

THR(6) The service(s) shall provide the capability to develop implementation plans.

THR(7) The service(s) shall provide the capability to develop initiation plans.

THR(8) The service(s) shall provide the capability to retrieve and view the history of items from implication scan through initiation.

THR(9) The service(s) shall provide the capability to tailor the process to fit organizational needs.

B. Objective Requirements

In addition to the threshold requirements, the author identified a few objective requirements that the architecture should support, even if the actual requirement was not supported in the first-pass implementation. These requirements follow in the list below.

OBJ(10) The service(s) shall provide search and retrieval capability of all historical data.
OBJ(11) The service(s) shall provide interfaces on portable devices.
OBJ(12) The service(s) shall provide Implication Scan submitters with the ability to interact with evaluators.
OBJ(13) The service(s) shall enable future capability growth.

C. Architecture

The requirements above specify an abstracted view of the desired application. They touch on the need for ubiquity and rapid feedback to stakeholders throughout the process. Service-oriented architecture provides a convenient method of enabling such ubiquity. Since the data repository itself can be realized as a service [2], client implementations are free to submit and retrieve data independently of the specific platform. Furthermore, Java standard edition applications, Web applications, and Java micro edition applications can be developed using the same service model, enabling the desired ubiquity. Once these applications are constructed, additional logic can be included in the service layer to monitor progress and enable rapid feedback. Finally, by moving the business logic into the services instead of the application, improvement and automation are possible without requiring client application upgrades.

The basic enabling technologies outlined above provide significant capabilities to a service-oriented application developer. As a framework, these technologies expose a broad collection of components that improve efficiency in adopting a service-oriented approach to an application. However, like all frameworks, their use requires design tradeoffs in the actual realization of the application. What serves well as a broad abstraction may limit what can be accomplished.

The PIR service-oriented architecture is shown in Figure 2. This architecture provides the capability to meet all of the threshold and objective requirements outlined above. The **PIR Services** process in the center represents the core collection of services. Services such as scan submission, current status, and backlog are realized through this interface. All consumers of PIR services are directed through the PIR Services endpoint. This includes both hand-held and desktop web services clients. Users accessing the service through a web browser also consume the PIR Services endpoint, but gain access through a server-side application (noted as the **PIR Web** process) that processes HTML requests.

The architecture accommodates a broad array of clients, from strategic planners at their desktop to individual contributors using a handheld[2]. Pairing of client and device places unique constraints on the realization of functionality. For example, a key difference between an enterprise edition Web service client (JSR-109) and a micro edition client (JSR-172 [20]) is that serialization and deserialization of objects via SOAP is limited under JSR-172. Hence, construction of the handheld PIR client places constraints on the services themselves. Where JSR-109 provides robust data type serialization, careful attention is required to ensure the service exposed for use by a Blackberry (JSR-172) can safely communicate with the device. The effect is handled in one of two ways: either objects are carefully constructed to use atomic data types that are supported by both standards; or custom logic is written to decompose complex objects into atomic types and reconstruct them after transmission. The PIR services require a combination of both, depending on whether the object can be cleanly represented as atomic types. As an example, textual elements within the database can be used as-is, but date-time stamps require special treatment.

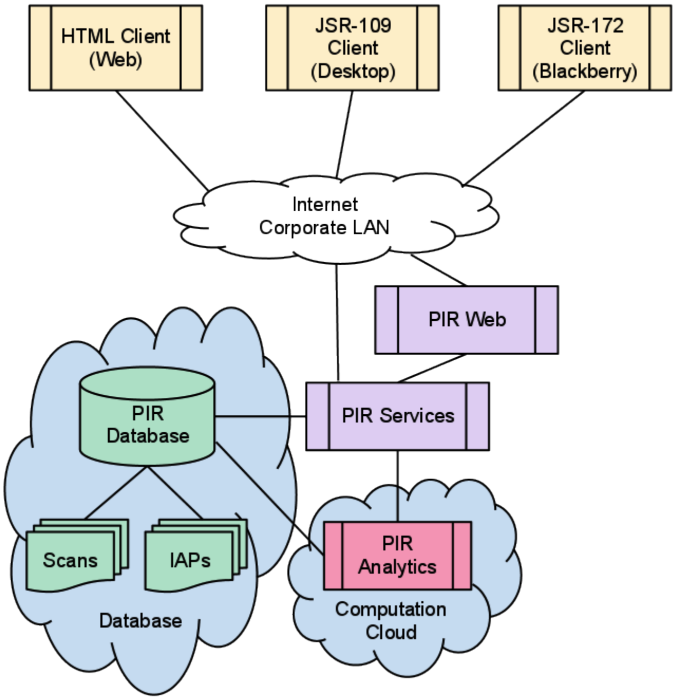


Fig. 2. PIR Architecture.

D. Database Considerations

The PIR services themselves are straightforward implementations in Java. Since the services are predominantly about exposing basic database functions, most service implementation code is trivial. JDBC provides an abstraction for many database backends and JPA provides an object-centric view of the database. When combined, basic storage and retrieval operations are reduced to a few lines of code. As an example, code for a prototype scan implementation is included below.

```
@WebMethod(operationName = "submitScan",
     action="submitScan")
public int submitScan(@WebParam(name = "trigger")
String trigger, @WebParam(name = "description")
String description) {
 EntityManagerFactory emf =
 Persistence.createEntityManagerFactory("PIR");
```

[2] Research In Motion's Blackberry Bold was chosen as a target handheld.

```
 EntityManager em = emf.createEntityManager();

 em.getTransaction().begin();

 People person = em.find(People.class, person);
 Implications impl = new Implications(0,
        new java.util.Date(), trigger, false);
 impl.setDescription(description);
 impl.setUserId(person);

 em.persist(impl);

 em.getTransaction().commit();

 em.close();
 emf.close();

 return impl.getId();
}
```

The JPA has abstracted away any notion of table retrieval or joins. Instead, table rows are object instances, and the relationships established between them are expressed naturally in the language. The API simplifies manipulations of the database, but there are some caveats to using it. Frequently tables within a database have a many-to-one or many-to-many relationships. Unchecked, these relationships have the potential to recursively enumerate large portions of the database in a single SOAP response. Frequently the solution is to identify when the persistence manager should ignore the relationship by annotating the relationship with the **@XmlTransient** tag. Doing so prevents the entity manager from enumerating the table rows specified by the relation.

JPA also raises a question about achieving the proper level of abstraction. The database backend, whether it is MySQL, Oracle, SQL Server, or some other implementation, is abstracted in Java using JDBC. JPA adds another layer above JDBC, providing an object-oriented view of the database and its relationships. One can argue that this additional abstraction is more of a hinderance to developers than a benefit [21]. Since SQL itself is intended to be an abstraction of the implementation, layers above the query language may actually obfuscate what is going on. Long term code maintainability by someone familiar with database manipulation may be better served by directly encoding the queries rather than relying on the JPA abstraction. Since JDBC already accommodates switching to a different backend with minimal impact, JPA's benefit may be minimal.

E. Security Considerations

Similar questions arise about security. While the WS-Security specification standardizes the mechanisms for securing web services messages, there is not a comprehensive web services security mechanism built into Java EE 6. Several standards have evolved that may be combined in support of a secure solution, but a comprehensive package for securing them does not exist. For example, certificate-based mechanisms incorporating timestamps provide the necessary strength, but exchange and management of certificates is difficult to administer [22]. The problem is compounded when developing a system where centralized identity management outside of the Java enterprise edition environment is essential. As of this writing, the author is still seeking an elegant solution to securing the web services that works well for both hand held applications and desktop environments.

F. Application Design

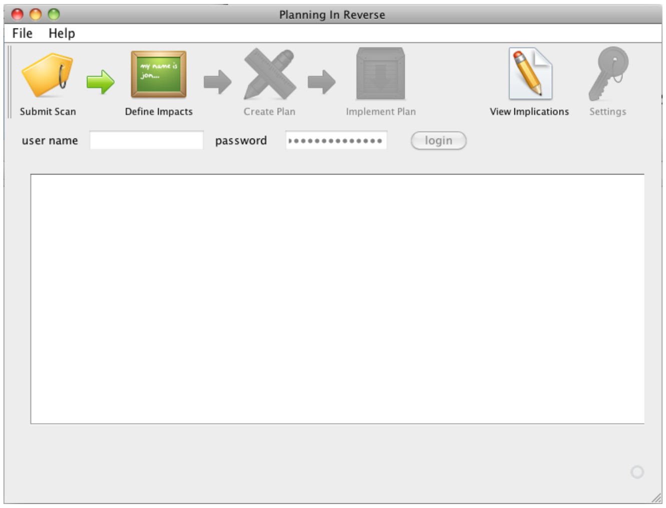


Fig. 3. PIR Desktop Client

.

Two primary client interfaces are required in the PIR implementation, one on a desktop the other on a handheld device. Though both rely on the same set of PIR services, their intended purposes are different. The handheld application is primarily used for implication scan submission and as a mechanism to track progress. As a result, the interface is quite simple, providing minimalist facilities to login, submit a scan (title and description), and check status. These three actions are collected in the application's menu.

The more complex application being developed allows strategic planners to manage the PIR process. The author elected to model the desktop-client user interaction as a process flow that follows the PIR workflow. Figure 3 shows the prototype desktop interface. The icons across the top represent steps in the PIR process. Like the handheld client, the desktop application uses the same core set of PIR services. Database access and business logic is abstracted by the services to provide a uniform interface.

G. Services

Below is a partial list of PIR functions decomposed into services. These services are currently incomplete, but the list provides insight into the basic functionality required. Perhaps the most significant benefit of an SOA implementation is the room for growth. As new functionality is identified that will improve the organization, it can be incorporated into the architecture with minimal impact.

submitScan: submit an Implication Scan for assessment.

getImplications: retrieve the contents of an Implication Scan.
submitImpact: submit an Impact Assessment for one or more Implication Scans.
getImpact: get an Impact Assessment.
getLateImplications: get Implication Scans that have aged too long.
closeImplication: close out an Implication Scan.
openImplication: reopen an Implication Scan.
closeImpact: close an Impact Assessment.
authenticate: authenticate a user for a session.

V. Conclusion and Future Work

The development of a Java-based Web services application provided significant opportunity to understand the facilities available under Java enterprise edition 6, and the progress made in facilitating service-oriented architecture development. It is clear that advances in technology, both from within the platform, and external to it in the development environment, have simplified the complexity associated with service-oriented architectures. Despite these advances, however, development of service-oriented architectures in Java still present some challenges. Security issues, platform compatibility, and API abstractions require analysis beyond simple design choices. Though the initial PIR application will remain in development for several more months prior to commercial use, this survey provided background in all of the technologies necessary to develop and deploy the final application.

The development and publication of new Web services provides ample opportunity for future expansion of the PIR application. Though the initial goal was simply to develop an information service that facilitated the use of the PIR process, future development efforts are likely to incorporate capabilities that become available as services. Both high-performance and cloud computing environments provide processing power necessary to perform complex artificial intelligence operations that may assist businesses in improving their strategic plans. The **PIR Analytics** box in Figure 2 encapsulates this expansion. Though these services may be undefined right now, it is clear that as computational capability increases, so too will its application to computationally difficult problems. In the future, strategic planning problems may be reduced to a specification that is transmitted to a service, processed, and returned back to PIR users. The elegance of a service-oriented architecture is that it is well positioned to embrace these potential capabilities without the need for substantial rework.

References

[1] S. Ballantyne, B. Berret, and M. E. Wells, *Planning In Reverse: A Viable Approach to Organizational Leadership*. New York: Rowman and Littlefield Education, 2011.

[2] A. Dan, R. Johnson, and A. Arsanjani, “Information as a service: Modeling and realization,” in *Systems Development in SOA Environments, 2007.*, May 2007, p. 2.

[3] D. Kafzig, K. Banke, and D. Slama, *Enterprise SOA: service-oriented architecture best practices*. Upper Saddle River, NJ: Pearson Education, 2005.

[4] S. Staab, W. van der Aalst, V. Benjamins, A. Sheth, J. Miller, C. Bussler, A. Maedche, D. Fesnel, and D. Gannon, “Web services: been there, done that?” *Intelligent Systems, IEEE*, vol. 18, no. 1, pp. 72–85, Jan-Feb 2003.

[5] T. Erl, “Service loose coupling,” 2009. [Online]. Available: http://www.soaprinciples.com/service_loose_coupling.php [Accessed: September 10, 2011]

[6] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web services description language (wsdl) 1.1,” March 2001. [Online]. Available: http://www.w3.org/TR/wsdl [Accessed: September 10, 2011]

[7] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon, “Soap version 1.2 part 1: Messaging framework,” April 2007. [Online]. Available: http://www.w3.org/TR/soap12-part1/ [Accessed: September 10, 2011]

[8] Oracle, “Your first cup: An introduction to the java ee platform,” March 2011. [Online]. Available: http://download.oracle.com/javaee/6/firstcup/doc/ [Accessed: September 10, 2011]

[9] E. Jendrock, I. Evans, D. Gollapudi, K. Haase, and C. Srivathsa, “The java ee 6 tutorial,” March 2011. [Online]. Available: http://download.oracle.com/javaee/6/tutorial/doc/ [Accessed: September 10, 2011]

[10] Oracle, “Glassfish server open source edition 3.1 application development guide,” April 2011. [Online]. Available: http://download.java.net/glassfish/3.1/release/glassfish-ose-3.1-docs-pdf.zip [Accessed: September 10, 2011]

[11] J. Kotamraju, “Web services for java ee, version 1.3,” December 2009. [Online]. Available: http://download.oracle.com/otn-pub/jcp/websvcs-1.3-mrel2-evaluate-oth-JSpec/websvcs-1_3-final-spec.pdf [Accessed: September 10, 2011]

[12] J. Ellis and M. Young, “J2me web services 1.0,” October 2003. [Online]. Available: http://download.oracle.com/otn-pub/jcp/j2me_web_services-1_0-fr-oth-JSpec/j2me_web_services-1_0-fr-spec.pdf [Accessed: September 10, 2011]

[13] K. Kawaguchi, S. Vajjhala, and J. Fialli, “The java architecture for xml binding (jaxb),” December 2010. [Online]. Available: http://download.oracle.com/otn-pub/jcp/jaxb-2.2-mrel2a-oth-JSpec/jaxb-2_2-mrel2-spec1.zip [Accessed: September 10, 2011]

[14] J. Kotamraju, “The java api for xml-based web services (jax-ws) 2.2,” December 2009. [Online]. Available: http://download.oracle.com/otn-pub/jcp/jaxws-2.2-mrel3-evalu-oth-JSpec/jaxws-2_2-mrel3-spec.pdf [Accessed: September 10, 2011]

[15] D. Williams, “Eclipse web tools platform project,” 2011. [Online]. Available: http://www.eclipse.org/projects/project_summary.php?projectid=webtools [Accessed: September 10, 2011]

[16] Oracle, “Netbeans ide 7.0 features: Web service development,” 2011. [Online]. Available: http://netbeans.org/features/web/web-services.html [Accessed: September 10, 2011]

[17] MySQL, “Mysql connectors,” 2011. [Online]. Available: http://www.mysql.com/products/connector/ [Accessed: September 10, 2011]

[18] L. DeMichiel, “Java persistence 2.0,” November 2009. [Online]. Available: http://download.oracle.com/otndocs/jcp/persistence-2.0-fr-eval-oth-JSpec/ [Accessed: September 10, 2011]

[19] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, “Web services security: Soap message security,” Feb 2006. [Online]. Available: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf [Accessed: September 10, 2011]

[20] BlackBerry, “Blackberry java application version 5.0 fundamentals guide,” April 2010. [Online]. Available: http://docs.blackberry.com/en/developers/deliverables/9091/JDE_5.0_FundamentalsGuide_Beta.pdf [Accessed: September 10, 2011]

[21] T. Neward, “The vietnam of compuer science,” Blog entry, June 2006. [Online]. Available: http://blogs.tedneward.com/2006/06/26/The+Vietnam+Of+Computer+Science.aspx [Accessed: September 10, 2011]

[22] A. Singhal, T. Winograd, and K. Scarfone, “Guide to secure web services: Recommendations of the national institute of standards and technology,” National Institute of Standards and Technology, Tech. Rep. 800-96, 2007.

Ontology-based Adaptive Reasoning Service for ScienceWeb

Hui Shi, Kurt J. Maly, and Steven J. Zeil
Department of Computer Science
Old Dominion University
Norfolk, VA
hshi/maly/zeil@cs.odu.edu

***Abstract*—ScienceWeb is a system that provides answers to qualitative and quantitative queries of a large knowledge base covering science research. The system will support the community joining together, sharing the insights of its members, to evolve the large knowledge base. ScienceWeb will need to scale to accommodate the substantial corpus of information about researchers, their projects and their publications. It will need to accommodate the inherent heterogeneity of both its information sources and of its user community. A reasoning system supports the queries and scalability becomes a serious challenge. In this paper we describe experiments and the resulting reasoning architecture and services whose scalability and efficiency are able to meet the requirements of query and answering in ScienceWeb. One key element of the services is an adaptive combination of both Query-invoked Inference and Materialization Inference together with incremental inferencing for more efficient query and answering. Second, we introduce new ways of storing, grouping and indexing objects in the knowledge base for faster searching and reasoning as the size of the triple set scales to the millions and the complexity of the Abox increases. The adaptive reasoning architecture and resulting adaptive reasoning service should provide efficient reasoning based on a scalable knowledge base.**

***Keywords*-knowledge bases; ontologies; reasoning**

I. Introduction

Consider a potential chemistry Ph.D. student who is trying to find out what the emerging areas are that have good academic job prospects. What are the schools and who are the professors doing groundbreaking research in this area? What are the good funded research projects in this area? Consider a faculty member who might ask, “Is my record good enough to be tenured at my school? At another school?” Similarly consider a National Science Foundation (NSF) program manager who would like to identify emerging research areas in mathematics that are not being currently supported by NSF. It is possible for these people each to mine this information from the Web. However, it may take a considerable effort and time, and even then the information may not be complete, may be partially incorrect, and would reflect an individual perspective for qualitative judgements. Thus, the efforts of the individuals neither take advantage of nor contribute to others’ efforts to reuse the data, the queries, and the methods used to find the data.

A number of projects (e.g., Arnetminer [1]) have built systems to capture limited aspects of community knowledge and to respond to semantic queries. However, these lack the level of community collaboration support that is required to build a knowledge base system that can evolve over time, both in terms of the knowledge it represents as well as the semantics involved in responding to qualitative questions. These systems are also homogeneous, in the sense that they harvest data from one type of resources. A team at ODU is working on ScienceWeb [2] [3], which will combine diverse resources such as digital libraries for published papers, curricula vitae from the web, and agency data bases such as NSF’s research grant data base and that will use collaboration as the fundamental approach to evolve its knowledge base.

Collaboration is at the heart of the approach to build ScienceWeb. Such collaboration includes building and evolving the knowledge base, building, evolving and reusing queries and identifying, specifying methods and harvesting raw information. The interaction between the system and people must be effective enough to allow for collaborative development.

Reasoning over the knowledge base provides support for answering qualitative questions and quantitative questions, whose scalability and efficiency influence greatly the response time of the system. ScienceWeb is a system that collects various research related information. The more complete the knowledge base is, the more helpful answers the system will provide. As the size of knowledge base increases, scalability becomes a challenge for the reasoning system. It may handle millions units of reasoning items in the knowledge base. As users make changes to the basic descriptors of the knowledge base, fast enough response time (ranges from seconds to a few minutes) in the face of changes is one of the core challenges for the reasoning system.

In this paper, we describe an adaptive reasoning architecture and an adaptive reasoning service whose scalability and efficiency are able to meet the interaction requirements in ScienceWeb system when facing a large and evolving knowledge base. The remainder of this paper is organized as follows: Section II describes the prior research relevant to this paper. Section III describes the work already done to implement parts of the ScienceWeb system. Section IV describes the adaptive reasoning service for ScienceWeb.

Section V presents the conclusions.

II. Background

There are a large number of knowledge bases for a variety of domains.

An example of a sophisticated knowledge base in Computer Science is Arnetminer [1], which provides profiles of researchers, associations between researchers, publications, co-author relationships, courses, and topic browsing. It has the capability to rank research and papers. It is a centrally developed system with fixed queries and schemas for data, but the knowledge base is continually growing as new data become available.

A. Ontologies

There has been increasing effort in organizing web information within knowledge bases using ontologies. Ontologies can model real world situations, can incorporate semantics that can be used to detect conflicts and resolve inconsistencies, and can be used together with a reasoning engine to infer new relations or proof statements. For example, the DBpedia [4] project focuses on converting Wikipedia [5] content into structured knowledge.

A number of tools exist for collaboratively designing and developing ontologies: for example, CO-Protégé [6], and Kaon2 [7].

Research in the field of knowledge representation and reasoning usually focused on methods for providing high-level descriptions of the world that can be effectively used to build knowledge-based systems. These knowledge-based systems are able to get implicit consequences of its explicitly represented knowledge. Thus, approaches to knowledge representation are crucial to the ability of finding inferred consequences [8]. Early knowledge representation methods such as frames [9] and semantic networks [10] lack well-defined syntax and a formal, unambiguous semantics, which are elements of qualified knowledge representation. Description Logic (DL) was therefore introduced into knowledge representation systems to improve the expressive power. Description Logic [11] is a family of logic-based knowledge representation formalisms, which is designed to represent the terminological knowledge from an application domain [12].

A DL knowledge base analogously consists of two parts: *intentional knowledge* (TBox), which represents general knowledge regarding a domain, and *extensional knowledge* (ABox), which represents a specific state of affairs. The "T" in the term "TBox" denotes terminology or taxonomy, which is built based on the properties of concepts and the subsumption relationships among the concepts in the knowledge. The "A" in the term "ABox" denotes assertional knowledge that includes individuals of the specific domain. [8] [13]

B. Inference Methods

The main reasoning tasks for DL reasoners are verifying KB consistency, checking concept satisfiability, concept subsumption and concept instances [12]. Algorithms for reasoning in DLs are: structural subsumption algorithms [14], the resolution-based approach [15], the automata-based approach [16] [17], and the tableau-based approach [18], which is currently the most widely used reasoning algorithm for DLs. There are three kinds of inference methods for First Order Logic (FOL): Forward chaining, Backward chaining and Resolution [19].

Materialization and Query-rewriting are the most popular inference strategies adopted by almost all of the state of the art ontology reasoning systems. *Materialization* means pre-computation and storage of inferred truths in a knowledge base, which is always executed during loading the data and combined with forward-chaining techniques. *Query-rewriting* means expanding the queries, which is always executed during answering the queries and combine with backward-chaining techniques.

Materialization and forward-chaining are suitable for frequent, expensive computation of answers with data that are relatively static. Owlim [20] [21], Oracle 11g [22], Minerva [23] and DLDB-OWL [24] all implement materialization during loading of the data. Materialization permits rapid answer to queries because all possible inferences have already been carried out. But any change in the ontology, instances, or custom rules requires complete re-processing before responding to any new queries. Furthermore, a large amount of redundant data may be produced by materialization of a large knowledge base, which may slow the subsequent loading and querying.

Query-rewriting and backward-chaining are suitable for efficient computation of answers with data that are dynamic and infrequent queries. Virtuoso [25] is one system that implements dynamical reasoning when it is necessary. This approach improves the performance of answering new queries after data changes and simplifies the maintenance of storage. But frequent repeated queries in query-rewriting will require repeated reasoning, which is time-consuming compared to pure search in materialization. Hstar [26] attempts to improve performance by adopting a strategy of partially materializing inference data instead of complete materializing.

A hybrid approach may give the best of both worlds. Jena [27] supports three ways of inferencing: forward-chaining, backward-chaining and a hybrid of these two methods. An adaptive hybrid approach would combine the strong points of both patterns for better performance under changing circumstances.

C. Storage Scheme

Semantic web applications can contain large amounts of data conformed to the ontology. How to store these large amounts of data and how to reason with them becomes a challenging.

Generally, there are two main kinds of ontology stores, native stores (disk-based stores) and database-based stores. Native stores are built on the file system, while database-based stores use relational or object relational databases as the back-end store.

Examples of native stores are OWLIM [20] [21], AllegroGraph [28], Sesame Native [29], Jena TDB [27], Hstar [26] and Virtuoso [25].

Representative database-based stores are: Jena SDB [27], Oracle 11g R2 [22], Minerva [23], (Sesame + MySQL) [29], DLDB-OWL [24] and (Sesame+ PostgreSQL) [29]. They all take advantage of existing mature database technologies for persistent storage.

The advantage of native stores is that they reduce the time for loading and updating data. However, a disadvantage of native stores is they are not able to make direct use of the query optimization features in database systems. Native stores need to implement the functionality of a relational database from the beginning, such as indexing, query optimization, and access control. As for database-based stores, the advantage is that they are able to make full use of mature database technologies, especially query optimization while the disadvantage is that they may be slower in loading and updating data,

D. Reasoning over Large ABoxes

Due to the large size of instance data conformed to corresponding ontology in many knowledge bases, reasoning over large ABoxes has become an issue in the fields of Semantic Web and Description Logics. There are two main kinds of approaches to dealing with this issue. The first approach includes designing novel algorithms, schemes and mechanisms that enhance the reasoning ability on large and expressive knowledge bases. Compared with some used state-of-the-art DL reasoners such as Racer, FaCT++, and Pellet, Kaon2 has been shown to have better performance on knowledge bases with large ABoxes but with simple TBoxes [30]. The second approach adopts simplification by reducing the expressive power of TBoxes describing large ABoxes. Calvanese et al. [31] have proposed a new Description Logic, called DL-Lite, which is not only rich enough to capture basic ontology languages, but also requires low complexity of reasoning.

We summarize the different ontology based reasoning systems along with the features they support in Table I.

III. ScienceWeb

This section describes the work we have already done to implement parts of the ScienceWeb system that includes the design of the architecture of ScienceWeb, the ontology in ScienceWeb and a synthetic data generator, the comparison of ontology reasoning systems, the study on a selected benchmarks and a formative study in Virginia.

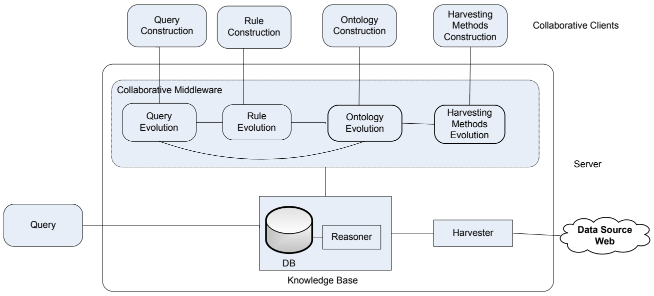


Figure 1. Architecture of ScienceWeb

A. Architecture

ScienceWeb is a platform where researchers including faculty, Ph.D. students and program managers can collaboratively work together to get answers of their queries from a consensus point of view or from their specific point of view. The collaborative aspect is not only in the construction of queries but in the construction of the underlying ontology, rules and instance data. The proposed architecture of the ScienceWeb is shown in Figure 1.

A traditional data mining architecture involves harvesting from data sources to populate a knowledge base, which in turn can then answer queries about the harvested content. We propose to enhance this architecture by adding a layer of collaborative clients for construction of queries, rules, ontological concepts, and harvesting methods, mediated by a layer of server functions that oversee and support the evolution of each of those functional groups within the knowledge base.

The system is built, developed and evolved based upon users' collaborative contributions. Users contribute during querying & answering, harvesting and ontology evolution. Querying is not an ordinary job of posting, parsing and retrieving as in a conventional database. Instead, it becomes an interactive, collaborative process. Harvesting and ontology evolution also benefit from the information provided by the users. Thus, collaboration is critical and widely spread throughout the system.

B. Performance of Existing Reasoning Systems

As described in Section II, there have been a number of studies on reasoning systems using only their native logic. To provide credibility for our context, we used benchmark data from these studies, replicate their results with native logic, and then extend them by adding customized rules. We used LUBM [32] for comparing our results with earlier studies. The second set of data will emulate a future ScienceWeb. Figure 2 shows the limited ontology class tree we deem sufficient to explore the scalability issues.

Both the LUBM and the ScienceWeb ontologies are about concepts and relationships in a research community. For instance, concepts such as `Faculty`, `Publication`, and

Table I
General comparison among ontology reasoning systems

	Supports RDF(S)?	Supports OWL?	Rule Language	Supports SPARQL Queries?	Repository: Persistent(P)/ In-Memory (M)
Jena	yes	yes	Jena Rules	yes	M
Pellet	yes	yes	SWRL	no	M
Kaon2	yes	yes	SWRL	yes	M
Oracle 11g	yes	yes	Owl Prime	no	P
OWLIM	yes	yes	Owl Horst	yes	P

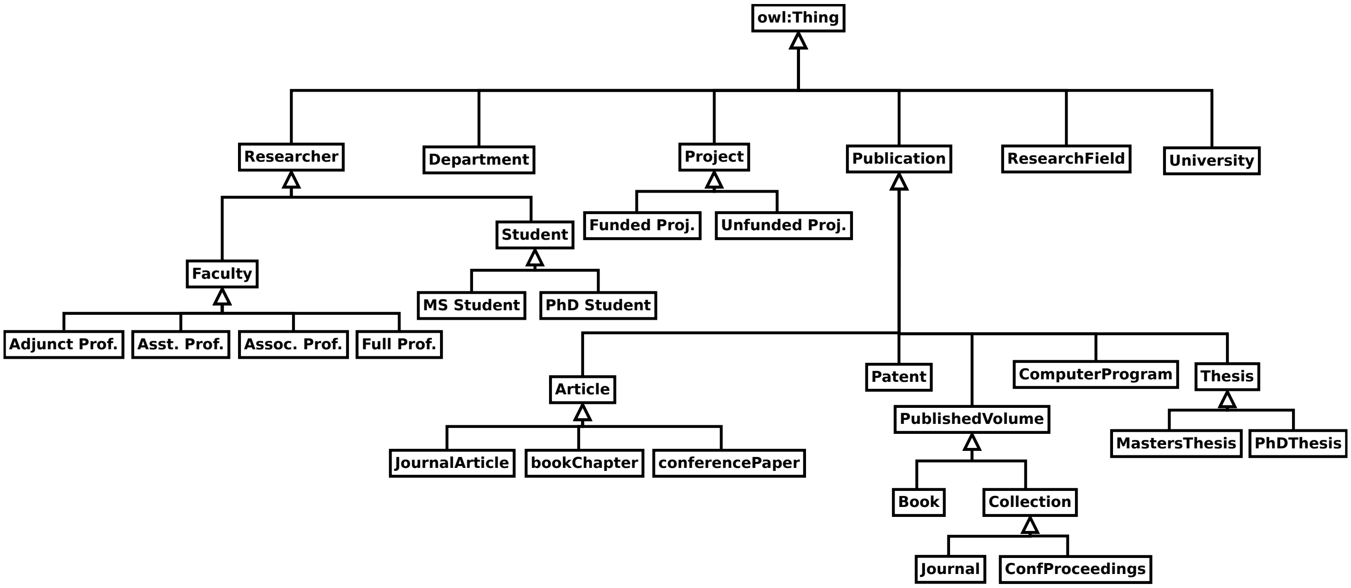


Figure 2. Class tree of research community ontology

`Organization` are included in both ontologies, as are properties such as `advisor`, `publicationAuthor`, and `worksFor`. All the concepts of LUBM can be found in the ScienceWeb ontology, albeit the exact name for classes and properties may not be same. ScienceWeb will provide more detail for some classes. For example, the ScienceWeb ontology has a finer granularity when it describes the classification and properties of Publication. The ontology shown in Figure 2 represents only a small subset of the one to be used for ScienceWeb. It was derived to be a minimal subset that is sufficient to answer a few select qualitative queries. The queries were selected to test the full capabilities of a reasoning system and to necessitate the addition of customized rules.

In support of the performance analysis described above, a flexible system for generating benchmark knowledge base instances of varying sizes has been developed [3]. The major challenge for this generator was to not only produce ontology-conformant data sets of the desired size, but to guarantee a plausible distribution for the many properties that relate objects across the knowledge base.

We performed a study to compare the scalability of existing reasoning systems when answering queries with customized rules in addition to native logic. We selected both LUBM and our own UnivGenerator [3] to provide the sample data and defined 5 custom rule sets for this experiment. Based on the comparison among these state-of-the-art ontology reasoning systems on full rule sets and transitive rule, we found [2] that OWLIM and Oracle offer the best scalability for the kinds of datasets anticipated for ScienceWeb, but they both have heavy loading and negative implications for evolving systems.

C. The Need for Agile Reasoning

The authors conducted a survey of faculty and PhD students at several universities in Virginia, aimed at determining both the demand for a ScienceWeb-like facility and the demands likely to be placed upon such a system [33]. Due to the similar faculty and student structure among universities in the USA, we believe this survey is likely representative of university communities in the USA. We have not yet explored differences likely to arise in a more international context. Notable results include:

- The perceived utility of a knowledge base with the proposed scope of ScienceWeb is high. Respondents indicated an interest in use of such information for both research and hiring/promotion purposes (including, apparently, job hunting by those expecting to acquire new PhDs). More than 90% of respondents indicated that they would have used such a system, had it been available, in the past year, to find information regarding hiring and promotion, with more than half believing they would have used such a system multiple times. Information on publications and projects had even higher perceived value, with 100% of respondents indicating that they would have used such a system multiple times in the past year.
- There is rather lower willingness to invest time in collaborative construction of such a system than might have been expected. Slightly over 40% of respondents would want to devote no more than a few minutes at a time to such activities, with another 30% willing to spend more than a few minutes but less than an hour.
  A notable exception to that rule is in updating and making corrections to one's own personal information in such a knowledge base, where more than 25% of respondents indicated they would likely devote multiple hours to validating and correcting information. This suggests that identification of low-effort collaborative techniques will be essential to the success of a ScienceWeb.
- Respondents generally believed themselves to share a consensus with both research peers and departmental colleagues on quality judgments pertaining to research. For example, more than 70% agreed with the statement "I share a consensus with my research peers on the criteria that define a groundbreaking researcher in my field."
  By contrast, they did not believe that they shared such a consensus on questions such as who might be good candidates for hiring or awarding of tenure, even with their Departmental colleagues. Less than 20% agreed with the statement "I share a consensus with the majority of my Departmental colleagues on the criteria for hiring a new assistant professor".
  This is an interesting distinction because one of the original motivations for ScienceWeb came from a desire for a trusted source of information to address disputes in the latter area.

One implication of these results is a system such as ScienceWeb must place a high premium on quick exploration of potential results. Users appear to be far more willing to wait for results to a "final" query than to engage in multiple iterations of time-consuming steps while composing a new query.

IV. Adaptive Reasoning Service for ScienceWeb

Although the technology for storing large knowledge bases is reasonably mature, the introduction of reasoning components into the query process poses a significant challenge. Our own preliminary study has shown that extant systems can provide fast responses to queries with substantial reasoning components, but only at a cost of extensive pre-processing, that is required to integrate new knowledge instances and new rules. In essence, there is a trade-off between scalability of query processing and the agility of the system in responding to changes in the supporting rules and evolution of the data.

A. Architecture

There are two competing strategies that can be used by this reasoning system: materialized inference, where all the rules in the system are fired at once and all inferred triples are generated, and query-invoked inference where relevant rules (TBox inference) in the system are fired after a query is accepted and partial inferencing (ABox inference) is done. To achieve the goal of improving the performance and scalability of reasoning, we propose an adaptive reasoning architecture that exploits both of these strategies. So far Jena is the only one system we found that contains both of these two strategies, but Jena's approach does not scale well to the size of knowledge base expected for ScienceWeb. The adaptive mechanism needs to determine what part of TBox inferencing is stable and what part of inferred instances should be temporary for incremental inferencing. For query-invoked inference the mechanism needs to determine what part of Tbox inferences should be pre-computed. As appropriate, the adaptive mechanism will switch between query-invoked inference and materialized inference. In addition, we introduce a long-term storage that will contain stable inferred instances from an incremental inference in materialized inference and pre-computed TBox inferences for a query-invoked inference. Any Abox results of a query-invoked inference will be saved in short-term storage. A query unaffected by changes to ontology, custom rules or instances can be answered by searching in the long-term storage. A query affected by changes to ontology, custom rules or instances can be answered by performing a query-invoked inference and searching the short-term storage.

The resulting architecture of an adaptive reasoning system is presented in Figure 3. The major modules comprising this architecture are as follows:

1) Input from users: A *query* is the basic way for users to search and retrieve information that they are interested. For example, "Who are the ground breaking researchers in Digital Libraries?" Here "ground breaking" is a qualitative descriptor that has been evolved by one (or more) user(s) by developing custom rules and researcher and digital library are classes in the ontology.

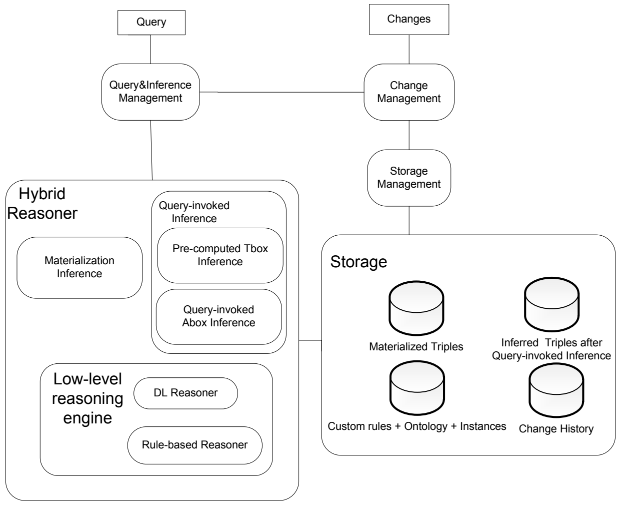

Query
Changes
Query&Inference Management
Change Management
Storage Management
Hybrid Reasoner
Query-invoked Inference
Materialization Inference
Pre-computed Tbox Inference
Query-invoked Abox Inference
Low-level reasoning engine
DL Reasoner
Rule-based Reasoner
Storage
Materialized Triples
Inferred Triples after Query-invoked Inference
Custom rules + Ontology + Instances
Change History

Figure 3. Architecture of an Adaptive Reasoning System

Changes refer to changes of ontology, custom rule set and instances as originally defined by users or harvested from the web. Generally, people might not agree on the ontology as it originally was designed and they will make changes according to their own beliefs. Similarly, custom rules represent a personal understanding of qualitative descriptors and as more people add their own opinion, they will change and, it is to be hoped, these qualitative descriptors will evolve to a consensus. The collection of instances will be enriched gradually with the discovery of new sources of information by individuals and the subsequent update of the methods of harvesting the information. Thus, changes of ontology, custom rule set and instances may occur at random or periodic times with varying degree of frequency. Changes have a significant influence on the process of storage and performance of query no matter whether the query involves inferencing or not.

2) Query & Inference Management: Query & Inference Management is the component that makes the choice between materialized inference and query-invoked inference. After a user submits a query, this component will determine if the query has been affected by the changes to the ontology, custom rules or instances by communicating with component of Change Management.

3) Change Management: Change Management maintains all the changes and arrangements in the Storage and records the change history after changes to the ontology, custom rules or instances. It not only provides change records to Query & Inference Management for the adaptive reasoning mechanism, but also communicates with the Storage Management module to realize the actual changes and operations in the storage.

4) Hybrid Reasoner: As a central component in the adaptive reasoning system, the Hybrid reasoner is a combination of Materialization Inference and Query-invoked Inference, and is responsible for the reasoning task with the assistance of a DL reasoner and a Rule-based reasoner.

Materialization Inference is one component that fires all of the rules in the system and generates all inferred triples at once. It implements all of the inference in advance following by various queries about the knowledge base from users. After Materialization Inference, answering query does not involve any reasoning but simple parsing and searching.

Query-invoked Inference is another component that invokes partial inference (ABox inference) when query is accepted after firing part of the rules (TBox inference) in the system in advance. Pre-computed TBox Inference is responsible for the TBox inference before queries while Query-invoked ABox Inference is responsible for the ABox inference during the query and answering.

5) Storage Management: The Storage Management module aims to update and to arrange the storage in an organized way to improve the scalability and performance of inferencing, especially the ABox inferencing. This component provides mechanisms to group and index base triples obtained from the users and the harvester module and triples that have been inferred such that search and updates can be done efficiently.

6) Storage: There are four separate storage areas of data in the system: Materialized Triples that are generated by Materialization Inference, Inferred Triples after Query-invoked Inference that are generated by Query-invoked Inference, Change History that is generated by Change Management and base storage including custom rules, ontology and instances as they were defined at startup of the system.

B. Adaptive Reasoning Mechanism

ScienceWeb is a collaborative platform that integrates efforts from users to define what data are to be obtained in what way and how the data are to be organized and what forms the queries will be. After a bootstrapping process has generated an initial knowledge base, we expect a period of frequent changes to all aspects of the knowledge base: ontology, rule set, harvesting methods, and instance data. It remains to be seen whether the ontology and rule set will stabilize over time. Instance data, however, will be continue to be changed via periodic harvesting.

The Adaptive Reasoning Mechanism is designed to select the appropriate reasoning method depending partially on the degree of change. Materialization inference is preferred in situations with infrequent or no updates. Any update of the ontology, custom rules or instances, however, would require re-loading of the data and re-materialization.

Query-invoked inference is preferred, therefore, in situations of rapid changes. As only related rules and data are involved, answers can be returned within an acceptable time period.

C. Knowledge base

Native storage of the knowledge base may help to improve the scalability of ABox inferencing. Native storage will speed up inferencing as well as search since it reducing the time for loading and updating data. Materialized Triples, Inferred Triples after Query-invoked Inference are stored separately from base storage (that including custom rules, ontology and instances) but we create a correlation index.

When we store triples, we group and index them to improve the inferencing performance. For example, triples can be grouped by each property, e.g., “publicationAuthor” or “advisor”. Triples with the same property can be stored in the same file. We will use Indexing as a way to store relationship and enhance the search performance.

D. Combination of DL and Rule-based Reasoners

DL reasoners have sufficient performance on complex TBox reasoning, but they do not have scalable query answering capabilities that are necessary in applications with large ABoxes. Rule-based OWL reasoners are based on the implementation of entailment rules in a rule engine. They have limited TBox reasoning completeness because they may not implement each entailment rule or they choose the performance instead of the completeness. Hybrid reasoners that integrate a DL reasoner and a rule engine can combine the strong points of both sides.

ScienceWeb introduces custom rules for description of qualitative and quantitative descriptors. Thus, besides the traditional Tbox and Abox reasonings, we introduce an Rbox reasoner for custom rules inferencing. The DL reasoner is responsible for TBox reasoning while the Rule-based Reasoner is responsible for both Rbox reasoning and ABox reasoning. These three kinds of reasoning are separable in the reasoning system for independent update of ontology, custom rules and instances.

These three kinds of reasoning are not independent. The DL reasoner generates specific entailment rules for the Rule-based Reasoner and ABox reasoning or Rbox reasoning is connected with TBox reasoning. We need to maintain correlations between individual objects.

V. Conclusion and Future Work

Our contributions in this paper are mainly in designing an adaptive reasoning architecture whose scalability and efficiency are able to meet the interaction requirements in ScienceWeb system, and in introducing the preliminary study that we have done for ScienceWeb system. With this architecture we are developing new technologies and an adaptive reasoning system for ScienceWeb that allows users to: (a) get answers effectively in real-time after posting existing qualitative queries, (b) get answers effectively in real-time after changing the ontology, custom rules, or instances and posting qualitative queries, and (c) get answers effectively in real-time when the size of knowledge base scales from thousands to millions.

Because reasoning is required for many large-scale semantic applications, we think it plausible that this architecture could be applied to a variety of different domains.

Sufficient reasoning support is a cornerstone for any realization of ScienceWeb. Our architecture and resulting system will, it is hoped, provide efficient reasoning based on a scalable knowledge base. We have completed the design, have performed a number of formative performance studies using a novel synthetic data generator, and have identified major issues we still need to address as:

Materialization inference - When can incremental inference be invoked? What part of TBox inference and inferred triples are stable for incremental inference?

Query-invoked inference - When can pre-computed TBox inferences be invoked? What pre-computed TBox inferences are? How to optimize query-invoked inference with large size of ABox?

Adaptive mechanism - When can the system switch between materialization inference and query-invoked inference? What are trivial changes and normal changes? Does the system need to differentiate the trivial changes and normal changes?

References

[1] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, “Arnetminer: Extraction and mining of academic social networks,” in *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD’2008).* ACM, 2008, pp. 990–998.

[2] H. Shi, K. Maly, S. Zeil, and M. Zubair, “Comparison of ontology reasoning systems using custom rules,” in *International Conference on Web Intelligence, Mining and Semantics*, Sogndal, Norway, 2011.

[3] A. Yaseen, K. J. Maly, S. J. Zeil, and M. Zubair, “Performance evaluation of Oracle semantic technologies with respect to user defined rules,” in *10th International Workshop on Web Semantics*, Toulouse, France, 2011.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “DBpedia-A crystallization point for the web of data,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.

[5] Wikipedia, *About Wikipedia — Wikipedia, The Free Encyclopedia*, 2010, http://en.wikipedia.org/wiki/Wikipedia:About [June 18, 2011].

[6] A. Diaz and G. Baldo, “Co-Protégé: A groupware tool for supporting collaborative ontology design with divergence,” in *The Eighth International Protégé Conference*, 2005, pp. 32–32.

[7] R. Volz, D. Oberle, S. Staab, and B. Motik, "Kaon server - a semantic web management system," in *Alternate Track Proceedings of the Twelfth International World Wide Web Conference, WWW2003*, Budapest, Hungary, 2003, pp. 20–24.

[8] D. Nardi and R. Brachman, "An introduction to description logics," in *The description logic handbook: theory, implementation, and applications*, 2002, pp. 1–40.

[9] M. Minsky, "A framework for representing knowledge," *The Psychology of Computer Vision*, 1975.

[10] M. Quillian, "Semantic memory," *Semantic Information Processing*, pp. 227–270, 1968.

[11] F. Baader, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge Univ Pr, 2003.

[12] F. Baader, I. Horrocks, and U. Sattler, "Description logics," *Foundations of Artificial Intelligence*, vol. 3, pp. 135–179, 2008.

[13] G. Stoilos, *An Introduction to Description Logics*, 2005.

[14] E. Mays, R. Dionne, and R. Weida, "K-Rep system overview," *ACM SIGART Bulletin*, vol. 2, no. 3, p. 97, 1991.

[15] Y. Kazakov and B. Motik, "A resolution-based decision procedure for SHOIQ," *Journal of Automated Reasoning*, vol. 40, no. 2, pp. 89–116, 2008.

[16] F. Baader, J. Hladik, C. Lutz, and F. Wolter, "From tableaux to automata for description logics," *Fundamenta Informaticae*, vol. 57, no. 2, pp. 247–279, 2003.

[17] C. Lutz, "Interval-based temporal reasoning with general TBoxes," in *Proc.of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, 2001, pp. 85–96.

[18] F. Baader and U. Sattler, "An overview of tableau algorithms for description logics," *Studia Logica*, vol. 69, no. 1, pp. 5–40, 2001.

[19] S. Russell and P. Norvig, *Artificial Intelligence: a Modern Approach*. Prentice hall, 2009.

[20] A. Kiryakov, D. Ognyanov, and D. Manov, "OWLIM - a pragmatic semantic repository for OWL," *Web Information Systems Engineering*, vol. 3807/2005, pp. 182–192, 2005.

[21] Ontotext, *OWLIM-OWL Semantic Repository*, 2011, http://www.ontotext.com/owlim/ [June 18, 2011].

[22] Oracle Corporation, *Oracle Database 11g R2*, 2011, http://www.oracledatabase11g.com [June 18, 2011].

[23] J. Zhou, L. Ma, Q. Liu, L. Zhang, Y. Yu, and Y. Pan, "Minerva: A scalable OWL ontology storage and inference system," *The Semantic Web-ASWC 2006*, pp. 429–443, 2006.

[24] J. Heflin and Z. Pan, "DLDB: Extending relational databases to support semantic web queries," in *Proceedings of Workshop on Practical and Scaleable Semantic Web Systems*, 2003, pp. 109–113.

[25] O. Erling, *Advances in Virtuoso RDF Triple Storage (Bitmap Indexing)*, June 2006, http://virtuoso.openlinksw.com-/dataspace/dav/wiki/Main/VOSBitmapIndexing [June 18, 2011].

[26] Y. Chen, J. Ou, Y. Jiang, and X. Meng, "HStar-a semantic repository for large scale OWL documents," *The Semantic Web-ASWC 2006*, pp. 415–428, 2006.

[27] Epimorphics Ltd, *Jena - a Semantic Web Framework for Java*, 2010, http://jena.sourceforge.net/ [June 18, 2011].

[28] Franz Inc, *AllegroGraph RDFStore 4.2.1*, 2011, http://www.franz.com/agraph/allegrograph/ [June 18, 2011].

[29] J. Broekstra, A. Kampman, and F. Van Harmelen, "Sesame: A generic architecture for storing and querying RDF and RDF schema," *The Semantic Web-ISWC 2002*, pp. 54–68, 2002.

[30] B. Motik and U. Sattler, "A comparison of reasoning techniques for querying large description logic aboxes," *Logic for Programming, Artificial Intelligence, and Reasoning*, vol. 4246/2006, pp. 227–241, 2006.

[31] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "Data complexity of query answering in description logics," in *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning*, 2006, pp. 260–270.

[32] Y. Guo, Z. Pan, and J. Heflin, "LUBM: A benchmark for OWL knowledge base systems," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, no. 2-3, pp. 158–182, 2005.

[33] K. Maly, S. Zeil, and M. Zubair, *ScienceWeb pre-survey*, 2010, http://www.cs.odu.edu/∼zeil/scienceWeb/survey2010/ [May 15, 2011].

Improving Games by SMS Through the MobileDeck Concept: A Quiz Game Proposal Focused on Emerging Markets

Mauro Ricardo da Silva Teófilo
Service Experience Department
Nokia Technology Institute
Manaus, Brazil
ext-mauro.teofilo@nokia.com

Marlon Farias da Luz
Service Experience Department
Nokia Technology Institute
Manaus, Brazil
ext-marlon.luz@nokia.com

Abstract—**This paper describes a case study of a quiz game designed to be used using SMS technology; the study consists of monitoring the game adaption to the MobileDeck concept. In the MobileDeck concept, the SMSs are received and sent through an appropriate graphical user interface. System efficiency and game improvement will be analyzed and discussed in this paper, in order to infer that the use of this proposed model is beneficial to the ecosystem of games based on SMS. The integration of the cited game to MobileDeck concept was proved a success, providing a new way of playing games via SMS.**

Keywords-mobile game; emerging market service; mobile application; SMS (Short Message Service)

I. Introduction

Short Message Service (SMS) has long been established as the de facto standard for sending and receiving text messages on mobile phones. As one of the most widely adopted communications services, it has been successfully used for over a decade as a marketing and services channel [1][2].

According to [1], the success of SMS communication may be attributed to 3 main reasons: it is ubiquitous, it has near real time delivery and it follows the “store and forward” mechanism, with later retransmission in case of failure. Moreover, the communication service holds a huge active users base [3], and a high response rate [4].

In emerging markets, those characteristics are especially crucial when designing mobile services, considering that mobile Internet is still gaining momentum [5]. In this scenario, text messaging is the most viable way to reach larger audiences, albeit compromising a considerable portion of the engagement effect [6][7].

Coulton et al. [8] conclude that although developing cellular applications for entertainment can be very challenging, it is also highly rewarding in that it provides the opportunity to produce new and exciting entertainment applications. Hence this paper makes use of the MobileDeck [9] concept to try improving the usability of games based on SMS exchange. MobileDeck is a complete solution, designed to improve the user experience of services that make use of SMS, adding a Graphical User Interface (GUI) to better represents the service. Thus, this implementation allows text messaging to behave exactly like a data feed to a rich application. In this sense, it offers a new, user-friendly channel for several existent SMS services like news, horoscope, promotions, and so forth [9].

The MobileDeck solution was launched in the Brazilian market in 2009 with various services, such as, location, weather forecast, news, etc., and by mid-2011 it had over 600,000 active users.

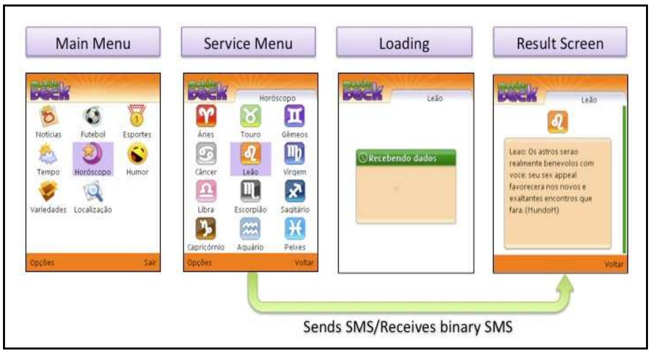
The purpose of this paper is to present a feasibility study of adding a game as a service to the MobileDeck solution. The game to be studied is a simple game in which the players attempt to answer questions correctly, like a quiz. In this case, the questions and alternatives are sent to the player via SMS. The interaction occurs when the user chooses an alternative, and sends it via SMS. This described game should be adapted to the MobileDeck concept, building a GUI to facilitate the interaction between players and game. Besides the general description of the technology used to enable it, this paper also aims to demonstrate the acceptance by potential users, establishing if this idea is valid for games adaptation or not.

II. MobileDeck

To better understand this study, it is necessary to introduce the MobileDeck concept. The key idea behind MobileDeck is to provide an attractive front-end for requesting and receiving content via SMS. On the client side, it consists of a mobile application capable of displaying both textual information and graphics using predefined layouts that are accessed through instructions contained in a binary SMS. In other words, every time the user requests a service, the application sends an SMS that is received by a specific server, and the response is again redirected to the application. The returning SMS acts like a script that builds and feeds the next screen with the appropriate content.

This solution can be explained through the example in Figure 1. From the main menu, which is a grid of icons, the user can choose one of the services, e.g. horoscope. That procedure activates the service screen (in this example, another grid menu). Once the desired content is chosen, the application requests the data by sending an SMS to a predefined short code. That request is then processed, while the application displays a “receiving data” feedback. When the response is received by the application (binary SMS), a result screen is populated with the respective data.

ISBN: 978-1-61208-152-6



Leao: Os astros serao realmente benevolos com voce: seu sex appeal favorecera nos novos e exaltantes encontros que fara. (MundoM)

Figure 1. Example of a service being accessed (horoscope) [9]

All the SMS traffic involved in this data request/receive process is thus transparent to the user. The response time is usually just a few seconds, so that the whole experience is extremely close to accessing content directly from the web using a mobile device. From that perspective, it is important to stress that this solution was designed for emerging markets, where fast data networks are still inaccessible to the majority of the population.

III. MobileDeck Architecture

MobileDeck is divided in four subsystems: a) a mobile client application, embedded into the mobile phone; b) a web based system, responsible for handling the requests and building a response message [10]; c) an aggregator that allows SMS traffic [11]; d) an information provider for each service.

The way each MobileDeck architecture component interacts is shown in Figure 2. This example illustrates the user requesting a service and receiving a response. First, an SMS is sent using the embedded application to a specific number using a standard protocol of the MobileDeck, then the SMS aggregator routes the message turning SMS into an HTTP request to the MobileDeck web server, which identifies the request and sends it to the appropriate information provider, which in turn returns the response content to it. The web server then builds a response using a security code and asks the aggregator to send a binary SMS with the return of service. At the end of the process the user receives the response from his service on the application embedded in his phone.

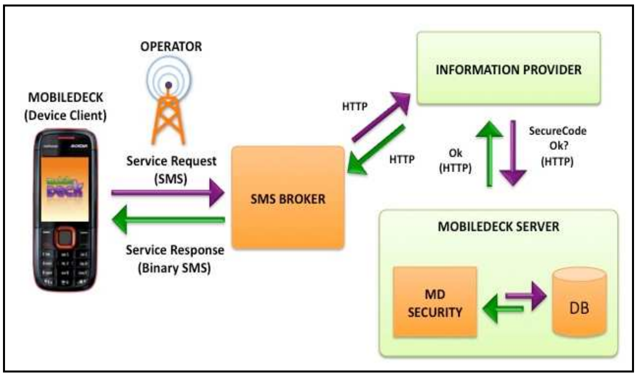


Figure 2. MobileDeck Architecture (high level design) [9]

The embedded application, designed for Nokia Series 40 and Series 60 devices, sends an SMS with a specified short code to achieve a determinate service. The service response is done by binary SMS, listing a predefined SMS port. The core programming language used to develop the mobile application was J2ME and its WMA 1.0 (Wireless Message API, JSR 120). The Series 60 version has a user interface developed with the Lightweight UI Toolkit (LWUIT), which is a free UI library and tool for creating richer and more portable Java ME user interfaces [12].

As MobileDeck sends protocol information inside a binary SMS body, a brief explanation of how it works is valid here. A binary SMS uses the port concept just like any Internet socket does. Thus, a message can be received in a given port that activates a specific service (in JME this is accomplished through the Push Registry API).

IV. Integrating a Quiz Game into MobileDeck

This section covers the process of integrating a set of questions and answers into the MobileDeck concept, creating a Quiz game by SMS with a rich user interface (UI).

This game of questions and answers, like a Quiz, has a version designed to be played by exchanging SMSs, with a good acceptance in the Brazilian market.

The game consists of receiving, via SMS, a text in the mobile phone’s inbox with a question and a list of alternatives. The player then sends by SMS what he believes to be the correct answer..

After receiving a positive feedback of the improved usability of SMS based services when they were adapted to the MobileDeck model, the decision was made to integrate and adapt the quiz game to the MobileDeck model, where the SMS exchanging is transparent to the user because he can interact with graphical elements that make up a representative screen of the service.

The first step was to add an icon to access the game in the main menu of MobileDeck.

The original game had to be adapted to the MobileDeck concept that offers the possibility of making the game more attractive to the user.

At the beginning of the game a list of categories from which the player can choose is displayed. The questions are based on the chosen category.

After choosing the category, the player chooses the name that will identify him/her in the game. The screen will contain a list of player names that have played using this application to facilitate the selection and an option to insert a new player name.

The name and category choice screens are illustrated in Figure 3(a) and (b), respectively.

After choosing the category and the player name a SMS with the data is sent that represents a request to start receiving questions, the SMS sending is transparent to the player and only messages like “sending request” and “waiting for question” are shown to the user.

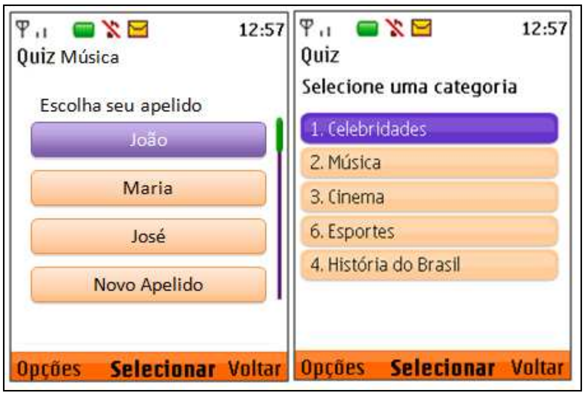


Figure 3. (a) Player name, and (b) Category choosing screen

The web server processes the sent question and, based on the category and player history, a binary SMS is sent following the "question screen" protocol, which contains all information to build properly the cited screen.

The embedded MobileDeck receives the binary SMS with all information needed to build a question screen, as illustrated in Figure 4. The question screen was designed to be used to choose an alternative and send the answer in a simple, intuitive and clear way.

When the player selects one of the alternatives on the question screen a SMS with the alternative chosen is sent to the web service which in turn processes it and defines a response depending on the fact if the player choose the right answer.

Together with the information about the last answer a new question is sent that is part of a set of questions to be answered by the player. At the end of the set, the score is shown as illustrated in Figure 5(a).

To make the game more attractive a ranking was created that displays the player names that have the highest score. Points are awarded to the player name every time he answers a question correctly. The ranking screen is shown in Figure 5(b).

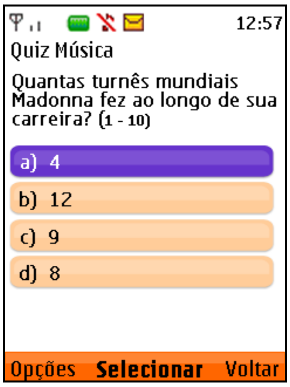
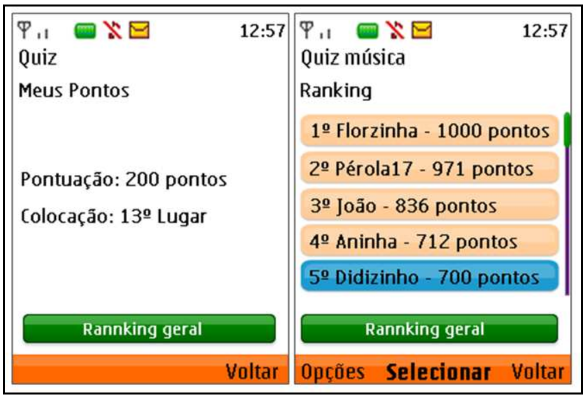

Quantas turnês mundiais Madonna fez ao longo de sua carreira? (1 - 10)

Figure 4. Question screen


Pontuação: 200 pontos

Colocação: 13º Lugar

1º Florzinha - 1000 pontos
2º Pérola17 - 971 pontos
3º João - 836 pontos
4º Aninha - 712 pontos
5º Didizinho - 700 pontos

Figure 5. (a) Score points of the player shown at the end of a set of questions, and (b) General ranking screen

V. Advantages and Disadvantages of Approaches

In this section, we aim to make a comparison between three approaches used to develop a mobile application which is described below. The idea of this application, which was created by [13], is to validate the creation of mobile entertainment applications.

The application was chosen in order to have a basis for comparison between existing services, a previously proposed service and the approach proposed in this paper.

The proposed application is based on the English Premier League, which is arguably one of the best football leagues in the world, and fans in the UK support it passionately.

The current service, which is in this section is called the first scenario, offers to mobile users work over the Short Message Service (SMS), and is generally limited to goal alerts, with the user paying by the number of SMS messages received. The application proposed by [13], which we call the second scenario, intends to improve both the timeliness of information and the range of information available. By providing information such as goals, red cards, yellow cards, substitutions, and so on, this application offers comprehensive updates on all the football matches on a given day. All the information is displayed in an easy accessible format, where a user can simply scroll through the event-by-event coverage of all the league games on that day.

Once all the events of the day become available to the application, the second part of this application becomes possible: a real-time fantasy football game. Fantasy football is played with a typical maximum budget of around £80 million for purchasing players. Players are selected from a list featuring over 500 footballers from Premiership clubs.

In the common approach, the user must use the native SMS editor to be able to order the SMS subscribe or order on demand. This approach is quite common and widely used; its main disadvantage is that to access the service you need to know its short number and an identification service code. Generally, media are used to achieve high penetration of the target audience, such as TV. Another disadvantage with this approach is that user interaction is extremely limited, with poor usability. Its main advantage is to work with a large user base and the adoption of this approach by many users generates massive user expertise.

The approach cited in Section 4 is called the third scenario.

In the approach that uses the MobileDeck concept working with a technology that is highly known and adopted is the great advantage over the second scenario approach, which uses GPRS technology to communicate with the information provider, thus limiting the number of potential users. When usability is concerned, the second and third scenario approaches, take a great leap in quality by adding a visual layer, as well as adding a range of options to create different kinds of applications.

The main disadvantage of the second and third scenarios is that for the user to be able to use these approaches, he needs to have an application installed on his mobile phone; the best solution to minimize this disadvantage is to have the application installed at the factory.

VI. Conclusion and Future Works

After completing game integration one could note that no feature of the questions and answers game designed originally for SMS was removed. New features such as name registers for identification and requests for overall rankings were added to the game. It is therefore possible to infer that the integration of the game to MobileDeck was proved a success in view of adaption and of providing a new way of playing games via SMS, where an attractive visual layer is added to enhance user interaction. Hence we can summarize that the main contribution of this paper is to demonstrate that the usability of games by SMS can be improved significantly through the use of the MobileDeck concept. As future work, a usability study should be conducted to try to measure the gain in usability by adopting the visual layer and whether the users of SMS games prefer to use the game via MobileDeck in case the game is brought to the emerging markets. If positive results are achieved, the integration of other games that can be played by SMS should be designed.

References

[1] Zerfos, P., Meng, X., Wong, S. H., Samanta, V., and Lu, S. 2006. A study of the short message service of a nationwide cellular network. In *Proceedings of* the 6th ACM SIGCOMM Conference on internet Measurement (Rio de Janeriro, Brazil, October 25 - 27, 2006). IMC '06. ACM, New York, NY, pp. 263-268. DOI= http://doi.acm.org/10.1145/1177080.1177114

[2] Hwu, J., Hsu, S., Lin, Y., and Chen, R. 2006. End-to-end security mechanisms for SMS. Int. J. Secur. Netw. 1, 3/4 (Dec. 2006), pp. 177-183. DOI= http://dx.doi.org/10.1504/IJSN.2006.011777

[3] Enck, W., Traynor, P., McDaniel, P., and La Porta, T. 2005. Exploiting open functionality in SMS-capable cellular networks. In Proceedings of the 12th ACM Conference on Computer and Communications Security (Alexandria, VA, USA, November 07 - 11, 2005). CCS '05. ACM, New York, NY, pp. 393-404. DOI= http://doi.acm.org/10.1145/1102120.1102171

[4] Eberspaecher, J., Bettstetter, C., and Vhogel, H. 2001 Gsm: Switching, Services and Protocols. 2nd. John Wiley & Sons, Inc.

[5] Tsang, M. M., Ho, S., and Liang, T. 2004. Consumer Attitudes Toward Mobile Advertising: An Empirical Study. Int. J. Electron. Commerce 8, 3 (Apr. 2004), pp.65-78.

[6] Kolko, B. E., Rose, E. J., and Johnson, E. J. 2007. Communication as information-seeking: the case for mobile social software for developing regions. In Proceedings of the 16th international Conference on World Wide Web (Banff, Alberta, Canada, May 08 - 12, 2007). WWW '07. ACM, New York, NY, pp. 863-872. DOI= http://doi.acm.org/10.1145/1242572.1242689

[7] Page, C. 2005. Mobile research strategies for a global market. Commun. ACM 48, 7 (Jul. 2005), pp. 42-48. DOI= http://doi.acm.org/10.1145/1070838.1070864

[8] Paul Coulton, Omer Rashid, Reuben Edwards, and Robert Thompson. 2005. Creating entertainment applications for cellular phones. Comput. Entertain. 3, 3 (July 2005), pp. 3-3. DOI=10.1145/1077246.1077254 http://doi.acm.org/10.1145/1077246.1077254

[9] Risi, D. and Teófilo, M. 2009. MobileDeck: turning SMS into a rich user experience. In Proceedings of the 6th international Conference on Mobile Technology, Application & Systems (Nice, France, September 02 - 04, 2009). Mobility '09. ACM, New York, NY, pp. 1-4. DOI= http://doi.acm.org/10.1145/1710035.1710068

[10] Pillai, P. 2005. Experimental mobile gateways. Crossroads 11, 4 (Aug. 2005), pp. 6-6. DOI= http://doi.acm.org/10.1145/1144389.1144395

[11] Brown, J., Shipman, B., and Vetter, R. 2007. SMS: The Short Message Service. Computer 40, 12 (Dec. 2007), pp. 106-110. DOI= http://dx.doi.org/10.1109/MC.2007.440

[12] White, Jim. Java ME User Interfaces: Do It with LWUIT. 2007. DevX online magazine. Retrieved from http://www.devx.com/wireless/Article/38461

[13] Paul Coulton, Omer Rashid, Reuben Edwards, and Robert Thompson. 2005. Creating entertainment applications for cellular phones. *Comput. Entertain.* 3, 3 (July 2005), pp. 3-3. DOI=10.1145/1077246.1077254 http://doi.acm.org/10.1145/1077246.1077254

An Approach of Early Disease Detection using CEP and SOA

Juan Boubeta-Puig, Guadalupe Ortiz, and Inmaculada Medina-Bulo
UCASE Software Engineering Group
Department of Computer Languages and Systems, University of Cádiz
Cádiz, Spain
Email: {juan.boubeta, guadalupe.ortiz, inmaculada.medina}@uca.es

***Abstract*—Service-Oriented Architectures (SOAs) have emerged as an efficient solution for modular system implementation, allowing easy communications among third-party applications; however, SOAs are not suitable for those systems which require real-time detection of significant or exceptional situations. In this regard, Complex Event Processing (CEP) techniques continuously process and correlate huge amounts of events allowing to detect and respond to changing business processes. In this paper, we propose the use of CEP in SOA scenarios to facilitate the efficient detection of relevant situations in heterogeneous information systems and we illustrate it through the implementation of a case study for detecting early outbreaks of avian influenza. Results confirm that CEP provides a suitable solution for the case study problem statement, significantly decreasing the amount of time taken to generate a warning alarm from the occurrence of an avian influenza outbreak and thus reducing disease impact.**

***Keywords*-CEP; complex event patterns; SOA 2.0; ESB; public health.**

I. Introduction

In recent years, Service-Oriented Architectures (SOAs) have emerged as an efficient solution for the implementation of systems in which modularity and communication among third parties are key factors. This fact has led to the increasing development of distributed applications made up of reusable and sharable components (services). These components have well-defined platform-independent interfaces, which allow SOA-based systems to quickly and easily adapt to changing business conditions. However, these architectures are not suitable for environments where it is necessary to continuously analyze all the information flowing through the system, which might be a key factor for an automatic and early detection of critical situations for the business in question.

This limitation may be solved by the joint use of Complex Event Processing (CEP) [1] together with SOA. CEP provides a set of techniques for helping to make an efficient use of Event-Driven Architecture (EDA), enabling it to react to multiple events under multiple logical conditions [2]. In this regard, CEP can process and analyze large amounts of events and correlate them to detect and respond to critical business situations in real time; in this scope event patterns are used to infer new more complex and meaningful events. These events will help to make decisions when necessary.

Currently, the integration of EDA and SOA is known as event-driven SOA (ED-SOA) or SOA 2.0 [3], an extension of SOA to respond to events that occur as a result of business processes. SOA 2.0 will ensure that services do not only exchange messages between them, but also publish events and receive event notifications from others. For this purpose, an Enterprise Service Bus (ESB) will be necessary to process, enrich and route messages between services of different applications. Thus, combining the use of CEP and SOA, we may detect relevant events in complex and heterogeneous systems, i.e., CEP will let us to analyze and correlate events in real time SOA 2.0.

To our knowledge, no architecture providing an appropriate and efficient integration of SOA, EDA, CEP and the detection of complex patterns has been proposed yet: there are proposals that use non-standard approaches to the integration of SOA and EDA [4], [5], while others use rule engines [6]. Implementations using rule engines are slower and less efficient in handling and receiving notifications, compared to those using CEP engines. Also, these approaches do not take into account that the system may have to handle a mass of events at any given time, causing a strong impact on system performance.

In this paper we propose an approach for the integration of SOA 2.0 and CEP in order to ease complex events detection in SOA scenarios. Showing the advantageous of using a CEP engine to facilitate an efficient detection of relevant situations is the main aim of this paper. Moreover, detection will be more efficient if the ESB prioritizes events by type, avoiding the bottleneck effect in the engine since CEP engine will analyze, firstly, higher priority events received at one particular point in time. In addition, this approach differs from others in the use of NoSQL (Not only SQL) databases [7], an emerging database management system based on key-value relationships, which is easily horizontal scalable and efficient for managing huge amounts of data.

In order to illustrate our proposal, a case study for detecting early epidemic outbreaks of diseases is also described in this paper. The case study will be implemented according the proposed technologies and will be evaluated through a simulation scenario.

The rest of the paper is organized as follows. In Section II

ISBN: 978-1-61208-152-6

we describe the main features of CEP and compare them with SOA's, it is followed by the proposed solution for their integration in Section III. Then, a case study for real-time detection of epidemic and pandemic cases of influenza is explained, implemented with the proposed technologies and tested in Section IV. Afterwards, in Section V our approach is discussed and in Section VI related approaches for the integration of CEP and SOA are summarized and compared to the one proposed in this paper. Finally, conclusions and future work are presented in Section VII.

II. CEP Background

CEP [1] is a technology that provides a set of techniques for helping to discover complex events by analyzing and correlating other basic and complex events. A basic event ocurrs at a point in time and it is indivisible and atomic, while a complex event can happen over a period of time, it is aggregated from basic or other complex events and contains more semantic meaning. Some of these techniques are: detecting causality, membership or timing relationships between events, abstracting event-driven processes and detecting event patterns. Therefore, CEP allows detecting complex and meaningful events, known as *situations*, and inferring valuable knowledge for end users.

The main advantage of using CEP to process complex events is that the latter can be identified and reported in real time, unlike in traditional software for event analysis, therefore reducing the latency in decision making.

Thus, CEP is a fundamental technology for applications that (1) must respond quickly to situations that change rapidly and asynchronously and where interactions do not have to be transactional, (2) must support management by exception, (3) must react rapidly to unusual situation and (4) require loose coupling and adaptability [8].

CEP has some similarities and differences with SOA. The main similarity is that both approaches provide modularity, loose coupling and flexibility. Some of the main differences are shown in the following lines:

- On the one hand, SOA interactions are based on services (a user must know the service producer and interface in advance in order to send requests to it). On the contrary, event-driven CEP is reactive and more decoupled since events are generated by event producers and consumers are responsible for intercepting and processing them.
- On the other hand, while SOA processes use events to drive control flow [9] (these processes can both send and receive events), CEP engines continuously analyze and correlate these events to assess if they meet the conditions defined in any of the event patterns stored in them.

III. Our Proposal in a Nutshell

We propose a solution based on the integration of CEP and SOA 2.0. A CEP engine is the key element of the integration, which will facilitate the efficient detection of relevant situations in heterogeneous information systems.

Event producers can be Web services, applications and sensors. Some of these applications are Web applications that allow users to interact with information management systems and legacy applications. Sensors are devices that monitor the environment to capture information (temperature, light, rain, etc.), which is then transmitted to the system using the controller integrated into the mentioned sensors.

These events are then published in the ESB and stored in a NoSQL database to be used as historical events.

Events are sent in parallel to the database management system for their storage as well as they are sent to event streams of a CEP engine. This engine will contain event patterns specifying the conditions to identify relevant situations and the actions to be carried out. Some of its functions are: filtering events (deleting irrelevant events from event streams), correlating and merging events from different event streams (complex events could be created) and aggregating them (grouping events). The generated complex event will be published immediately into the ESB.

Finally, these events will be notified to the event consumers that have subscribed to them. These consumers can be Web services, applications (such as dashboards for displaying alarms) and/or actuators that perform some action (switch on/off, open/close, etc.) on a specific device.

IV. Case Study

In recent decades the globalization has caused a huge increase of people movements between countries resulting in a dramatic increase of the impact of emerging disease epidemics. This situation is becoming a major threat to life, safety and the world economy [10]. This fact motivated the decision of implementing a case study to detect avian influenza outbreaks in real time.

Thus, the objective of this case study is to demonstrate that CEP is an effective solution for detecting epidemics and pandemics in real time compared to most existent tools that report these situations weekly: FluNet [11] presents influenza information in all countries of the world and Euroflu [12] presents it only in member states of the WHO European region. The use of a CEP engine will allow health officials to mitigate as soon as possible the impact of epidemics and global pandemics, rather than being exposed to have a week delay on receiving the up-to-date information.

In the following subsections we describe the case study, define the complex event patterns necessary to detect critical situations in this scenario, enumerate the steps followed to implement the case study and finally present the results obtained after testing it.

A. Description of the Case Study

As previously mentioned this case study focuses on early detection of avian influenza outbreaks using CEP in SOA environments. In particular, this health system will launch real-time alerts when some of the following avian influenza cases are detected: (1) suspected cases of patients who may be infected by this virus, (2) confirmed cases of suspected patients, (3) epidemic cases (countries suffering outbreaks of avian influenza) and (4) pandemic case (the epidemic affects several countries). See WHO documentation [13] for further information about the definition of these cases.

The event producers are:

- **Hospitals**: health workers, particularly physicians, will diagnose symptoms and will get relevant information about patients. They will issue events inserting patient diagnoses in hospital information systems.
- **Laboratories**: laboratories will be able to detect confirmed cases of avian influenza by blood tests and other techniques, and they will publish this information as events within their information systems.

On the other hand, the event consumers are:

- **WHO and other international organizations**: these organizations will subscribe to relevant events about outbreaks of avian influenza and will be aware of suspected, confirmed, epidemic and pandemic cases that might have been detected worldwide.
- **Hospitals**: health workers will need to know which cases have been identified in order to take measures for relieving the situation, such as patient isolation measures.
- **Laboratories**: they will be continuously informed of the virus evolution and spread, facing the development of new antidotes or drugs to help authorities to fight the disease.

For example, hospital services may trigger a complex event if a suspected case of avian influenza is detected. This event will be received by the pharmacy and WHO services, which will react immediately to this situation: pharmacies automatically notify to suppliers an increased demand for those drugs that help fighting the disease and WHO will launch warning alarms to those laboratories and international health agencies that are interested in this situation.

B. Complex Event Patterns for Detecting Avian Influenza Outbreaks

In the following lines, we describe the definition of complex event patterns for detecting suspected, confirmed, epidemic and pandemic cases of avian influenza. To this end, we have adapted Buschmann's design patterns scheme [14]: pattern *name* and a short summary, real-world *example* demonstrating the existence of the problem and the need for the pattern, *context* (situations) in which the pattern may be applied), *problem* addressed by the pattern, *solution* proposed by the pattern, detailed specification of the pattern *structural aspects*, pattern *implementation* in a specific language and *consequences* (benefits and drawbacks) provided by the pattern. In this work we will describe the pattern name and implementation, which are the main relevant parts of the schema for the case study illustration.

According to real requirements for detecting avian influenza cases we defined the next complex event patterns:

- **Suspected case**: this pattern detects possible occurrences of avian influenza cases, when the following conditions are met:
  1) The patient has fever (above 38 °C) or cough or headache or myalgia or conjunctivitis or pharyngitis or encephalopathy or multiple organ failure or pneumonia.
  2) And, moreover, he/she presents a history of exposure to known infection sources in infectious period (7 days prior):
     – Staying in an area where avian influenza human cases have been reported.
     – Having contact with a person already diagnosed of avian influenza.
     – Having contact with animals that could be infected.
     – Handling gases in a laboratory.
- **Confirmed case**: the laboratory confirms an avian influenza infection, based on the detection of a suspected case and a biological sample of the patient.
- **Epidemic case**: there are 25 or more confirmed cases of avian influenza in a particular country during a week.
- **Pandemic case**: there are 2 or more epidemic cases during a week.

C. Implementation

The presented case study has been implemented using Java and the Esper engine. Moreover, the complex event patterns defined above have been implemented in the complex event processing language of Esper [15], EPL (Event Processing Language). Several reasons have motivated EPL choice: firstly, the learning curve is not high because its syntax is very close to SQL, widely known worldwide. Besides, EPL natively supports multiple event format types: Java/.NET objects, maps and XML documents what facilitates its use in multiple platforms. Even more, it is also possible to customize not only the language but also Esper engine, which is written in Java and is open source.

The steps followed to implement the case study are enumerated and described below:

1) **Configuration and initialization of the Esper engine**. An instance of *com.espertech.esper.client.Configuration* represents all configuration parameters. The *Configuration* is used to build an *EPServiceProvider*, which provides the administrative and runtime interfaces for an Esper

engine instance. A *Configuration* instance is then obtained by instantiating it directly and adding or setting values on it. The *Configuration* instance is then passed to *EPServiceProviderManager* to obtain a configured engine, as the following code shows:

```
Configuration conf = new Configuration();
config.addEventType("PatientState",
  PatientState.class.getName());
config.addImport("es.uca.esper");
EPServiceProvider epService
  = EPServiceProviderManager.getProvider(
  "sample", conf);
```

2) **Creation of an event generator to simulate patients treated worldwide and their health state evolution**; the simulator will be directly connected to the CEP engine. We will make use of this simulator to produce patient state events randomly rather than using real information from hospital and laboratory systems (due to the access restrictions to official information in the above mentioned systems). In this simulator there are two types of objects:
   - *Patient*: each patient in the simulation has a specified person id, date of birth, sex and country.
   - *PatientState*: this object represents patient health state evolution, which has the following attributes: identification, registration time, current location of the person, symptoms and dates in which the patient has been exposed to infection sources.

3) **Introduction of the generated events in Esper event streams**. The *PatientState* instance insertion in Esper is implemented as follows:

```
epService.getEPRuntime().sendEvent(
  PatientState);
```

4) **Implementation and registration of complex event patterns in Esper**. The suspected case of avian influenza implemented using EPL is presented below:

```
String suspectedCase =
"insert into AvianInfluenzaSuspects
select avianInfluenzaSuspect.id,
  avianInfluenzaSuspect.registrationTime,
  avianInfluenzaSuspect.patient.sex,
  avianInfluenzaSuspect.currentLocation
from pattern [every avianInfluenzaSuspect
  = PatientState((cough or fever > 38
  or headache or multipleOrganFailure
  or myalgia or pharyngitis or pneumonia
  or conjunctivitis or encephalopathy)
  and ((PatientState.DayCounter(
  registrationTime, infectionArea)<=7)
  or (PatientState.DayCounter(
  registrationTime, infectionPerson)<=7)
  or (PatientState.DayCounter(
  registrationTime, infectionAnimal)<=7)
  or (PatientState.DayCounter(
  registrationTime, laboratoryGases)<=7)
)]";
EPStatement suspectedCaseStatement =
  epService.getEPAdministrator().
  createEPL(suspectedCase);
suspectedCaseStatement.addListener(
  new AvianInfluenzaSuspectListener());
```

Concerning the code, the complex event pattern illustrating suspected case implementation is defined by the *from pattern* clause. *PatientState* events meeting described conditions for suspect case are selected from the Esper event stream. For this purpose, *every* operator is applied to obtain all these events and the *avianInfluenzaSuspect* alias is assigned to them.

DayCounter is a function to count days passed from the date on which *PatientState* event was registered to the date on which the patient was in contact with a risk source, if there were any contact.

Afterwards, identification, registration time, sex and current location attributes of the met *avianInfluenzaSuspect* complex events are selected and inserted in a new event stream called *AvianInfluenzaSuspects*, by using an *insert into* clause.

A specific listener, known as *AvianInfluenzaSuspectListener*, will receive suspect patient event notifications and will alert those interested to these situations. These warning alarms could be used to infer statistical data, e.g., the amount of suspected case grouped by sex in a specific time for a given country.

5) **Detection of complex events according to the registered patterns and notification of these events to the listeners**. The implementation of *AvianInfluenzaSuspectListener*, which receives events detected by *SuspectedCase* pattern, is shown below:

```
public class AvianInfluenzaSuspectListener
  implements UpdateListener {
    @Override
    public void update(EventBean[]
      newEvents, EventBean[] oldEvents) {
        String currentLocation =
          (String) newEvents[0].
           get("currentLocation");
          System.out.println(
           "\n***SUSPECTED CASE IN: " +
           currentLocation + "***\n"); }}
```

6) **Definition and implementation of test cases using the JUnit framework and validation of the application**. For example, the *testGen* test case is presented below, which checks if our implemented event generator creates the specific amount of events and insert them in an event stream:

```
public void testGen() throws Exception {
    final int EVENT_N = 100000;
    PacientStateGenerator generator =
      new PacientStateGenerator();
    LinkedList stream = generator.
      makeEventStream(EVENT_N);
    assertEquals("The amount of events
      generated randomly should be " +
      EVENT_N, stream.size(), EVENT_N); }
```

D. Testing and Results

In this study, 100.000 - 600.000 patient states, from 119 countries, have been generate randomly, using the implemented event generator.

In our simulations we have observed that up to 300.000 generated patient states no epidemic case has been alerted. The reason is there are not enough patient states to detect suspected cases, which are also confirmed, requiring at least 25 confirmed cases for the same country during a week.

Only 2 epidemic cases have been detected of 350.000 patient states. However, the amount of epidemic cases significantly increased from 400.000 states. We can deduce that the more patient states are generated the more epidemic cases will be detected.

As a conclusion, we can assert that using CEP permits an inmediate and efficient detection of complex patterns in large amounts of flowing information.

V. Discussion

Through the case study implementation and evaluation we have seen that our proposal provides an efficient solution for early detection of avian influenza outbreaks. Besides we can stress the following additional advantageous characteristics:

The use of a CEP engine instead of a rule engine provides substantial benefits, as discussed in this section. According to Chandy and Schulte [8] there are some differences between CEP and rule engines: normally, rule engines are request-driven, i.e., when an application needs to make a decision it will invoke this engine to derive a conclusion from a set of premises. The general model for a rule engine is *If "some condition" then "do action X"*. In most applications, a large number of rules will have to be analyzed before making a decision, thus becoming a problem for real-time decision making. However, CEP engines are event-driven and run continuously, and according EDA principles, they can process notification messages as soon as they arrive. In this case, the general model for a CEP engine is a when-then rule (known as a *complex event pattern*) *When "something happens or some condition is detected" then "do action X"*, instead of an if-then-else rule. The equivalence of the if-else clause for a CEP engine is the one that specifies *When "something has not happened in a specific time frame" then "do action Y"*. Thus, event patterns use time as another dimension. Moreover, CEP engines are faster and more efficient in handling and receiving notifications since they can directly manage inputs and outputs with messaging systems, while rule engines behave as services used by input and output systems.

Another improvement is our approach provides event prioritization according to the order previously set for every event type. Event prioritization will prevent the bottleneck effect in the CEP engine, as it will serve firstly higher priority events, thus avoiding the management of a huge number of events at one particular point in time.

Finally, our proposal uses NoSQL databases. They provide the following adavantages [16]:

- NoSQL have asynchronous BASE (Basically Available, Soft state, Eventual Consistency) updates rather than synchronous ACID (Atomicity, Consistency, Isolation, Durability).
- NoSQL databases are optimized to react to changes, not to manage transactions, and they do not require neither schemes nor data types definitions.
- They are also distributed, easily horizontal scalable and very efficient for managing huge data amounts.

VI. Related Work

Several works about CEP and SOA integration in different domains can be found in the literature; in the following paragraphs we summarize the most representative ones.

To start with, He et al. [4] implement an event-driven system based on radio-frequency identification to monitor gases emitted by vehicles and detect if vehicle's emissions are not standard, keeping those interested in protecting the environment and quality air informed about this situation. The authors claim that all events in the CEP engine are represented by POJO (Plain Old Java Object); the language used to process events is similar to SQL, however it is not specified whether this language has been extended to manipulate time windows, which is a relevant feature for a CEP application. Besides, events are not represented in XML (what would allow obtaining more readable and reusable events) as the Esper engine [15] does. On the other hand, they do not specify whether the motor and the language can be customized and extended as Esper allows to.

On the other hand, Taher et al. [5] propose to adapt interactions of Web service messages between incompatible interfaces. In this regard, they develop an architecture that integrates a CEP engine and input/output adapters for SOAP messages. Input adapters receive messages sent by Web services, transform them to the appropriate representation to be manipulated by the CEP engine and send them to the latter. Similarly, output adapters receive events from the engine, transform them to SOAP messages and then they are sent to Web services. This architecture has some limitations regarding our proposal: firstly, services interact with a framework that integrates both message adapters and the CEP engine, instead of using an ESB directly connected to the services and the engine. The bus would provide a decoupled, flexible and reusable system. Secondly, the proposed adapters do not provide additional functionalities which an ESB usually provides, such as message routing based on content and transformation protocols.

Sottara et al. [6] propose an architecture for the management of waste water treatment plants, in which an ESB is used to connect services distributed in different nodes in a transparent way for end users. They integrate JBossESB solution [17] and Drools rule engine [18], which is embedded in the bus. Our proposal improves this one by using a CEP engine instead of a rule-based one.

Finally, there are two projects funded under the EU 7th Research Framework Programme that integrate CEP and SOA: MASTER and COMPAS. MASTER [19] provides an infrastructure that facilitates monitoring, enforcement, and auditing of security compliance and COMPAS [20] designs and implements an architectural framework to ensure dynamic and on-going compliance of software services to business regulations and stated user service-requirements. However, while our solution allows to store event logs in NoSQL databases, these projects do not consider it.

VII. Conclusion and Future Work

We have proposed and discussed an approach for the efficient use of CEP in SOA 2.0 scope. Thanks to this approach, when relevant situations arise from the detection of certain predefined event patterns, real-time alerts will be sent to the interested parties. In this paper, we have mainly focused on the use of a CEP engine to detect event patterns.

A case study illustrating this approach has also been described and implemented. Our system can detect epidemics and pandemics in real time, while most current tools report these situations weekly. So, we can conclude that CEP technology is suitable for this purpose. Although we have taken the example of the avian influenza virus; once new complex event patterns are defined, our system could be used for the prevention of other diseases as well as for non-medical fields, should it be necessary.

In our near future work we will approach a complete architecture for the integration of SOA 2.0 and CEP making use of an ESB. In this regard, the CEP engine will be able to process *real* events that will be published into the ESB by different event producers, replacing the random event generator developed in this work to simulate patient states. Moreover, both event producers and consumers will be Web services, providing a loosely coupled more complex, modular and flexible system.

Acknowledgements

We would like to specially thank Novayre [21] managers for their fruitful comments and discussions on the topic dealt with in this paper. The second author acknowledges the support from TIN2008-02985 and FEDER.

References

[1] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. MA, USA: Addison-Wesley, 2002.

[2] H. Taylor, A. Yochem, Les Phillips, and F. Martinez, *Event-Driven Architecture: How SOA Enables the Real-Time Enterprise*. Indiana, USA: Addison-Wesley, Mar. 2009.

[3] B. Sosinsky, *Cloud Computing Bible*. Indiana, USA: Wiley, Jan. 2011.

[4] M. He, Z. Zheng, G. Xue, and X. Du, "Event Driven RFID Based Exhaust Gas Detection Services Oriented System Research," in *Proc. 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, Oct. 2008, pp. 1–4.

[5] Y. Taher, M. Fauvet, M. Dumas, and D. Benslimane, "Using CEP Technology to Adapt Messages Exchanged by Web Services," in *Proc. 17th International Conference on World Wide Web*, Beijing, China, Apr. 2008, pp. 1231–1232.

[6] D. Sottara, A. Manservisi, P. Mello, G. Colombini, and L. Luccarini, "A CEP-based SOA for the Management of WasteWater Treatment Plants," in *Proc. IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems*, Crema, Italy, Sep. 2009, pp. 58–65.

[7] "NoSQL databases," Mar. 2011. [Online]. Available: http://nosql-database.org/

[8] K. M. Chandy and W. R. Schulte, *Event Processing: Designing IT Systems for Agile Companies*. USA: McGraw-Hill, 2010.

[9] M. Havey, "CEP and SOA: Six Letters Are Better than Three," Feb. 2011. [Online]. Available: http://www.packtpub.com/article/

[10] "United Nations," Mar. 2011. [Online]. Available: http://www.un.org/en/

[11] "FluNet," Apr. 2011. [Online]. Available: http://www.who.int/csr/disease/influenza/influenzanetwork/flunet/en/

[12] "EuroFlu," Apr. 2011. [Online]. Available: http://www.euroflu.org/index.php

[13] "World Health Organization," Mar. 2011. [Online]. Available: http://www.who.int/en/index.html

[14] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns*. Chichester, UK: Wiley, 1996.

[15] "Esper," Mar. 2011. [Online]. Available: http://esper.codehaus.org

[16] E. Meijer and G. Bierman, "A co-Relational Model of Data for Large Shared Data Banks," *ACM Queue*, vol. 9, pp. 30–48, Mar. 2011.

[17] "JBoss ESB," Jan. 2011. [Online]. Available: http://jboss.org/jbossesb

[18] "Drools," Jan. 2011. [Online]. Available: http://www.jboss.org/drools

[19] "MASTER," May 2011. [Online]. Available: http://www.master-fp7.eu/

[20] "COMPAS," May 2011. [Online]. Available: http://www.compas-ict.eu/

[21] "Novayre," May 2011. [Online]. Available: http://www.novayre.com/

The Deployment of Service Management Systems in SMEs – Three Case Studies

Martin Wynn
Department of Computing
University of Gloucestershire
United Kingdom
mwynn@glos.ac.uk

Emma Tipton
Department of Computing
University of Gloucestershire
United Kingdom
e.tipton@muddyboots.com

***Abstract*—Service management is a new and emerging software niche, which is still relatively unexplored by many companies. This paper examines the service management process in three small to medium sized enterprises in the UK Midlands region, and the software systems that support the process. The paper provides a top-line analysis of the degree of fit between process and technology support, and concludes with some observations on service management as a process and systems concept.**

Keywords - service management, SMEs, information systems, process change, case studies

I. INTRODUCTION

Service management is a relatively new concept as regards business information systems. Only recently has it emerged as a separate software module, either as a specialist standalone offering (such as Solavista) or as a module in an integrated ERP package, such as SAP. Service management can be defined as "monitoring and optimizing a service to ensure that it meets the critical outcomes the customer values and stakeholders want to provide" [1]. Service management has also been seen as "comprised of software, services and knowledge that assist companies in efficiently delivering service commitments" [2]. Software vendors suggest that systems can cover a range of functions related to service scheduling, delivery and monitoring. One software provider asserts that service management systems are "designed for companies that market, sell, service and support equipment and/or assets or provide professional services. They start working from the point of initial lead/opportunity, through quotation/estimate, order processing, installation/ commissioning, servicing and maintenance to equipment/asset retirement and replacement" [3].

Before the advent of these software packages, systems users either had to adapt other packaged software to embrace the required functions as best they could, or bespoke in-house systems to cover these requirements. One such example of the latter was the Keg Information Management system (KIMS) developed at cider-maker HP Bulmer in the early 1990s. The company was migrating its systems to the Oracle ERP product, but this software, from one of the World's leading suppliers, did not provide the required functionality to support the scheduling and installation of bar top cider fonts and cider dispense equipment in the pubs and clubs around the country. This early example of a bespoke service management system exhibited many of the features of later packaged solutions. Bloor [4] noted in 1993:

"The keg equipment installed in these outlets represents a major investment for Bulmer, and its installation and repair is the responsibility of some 40 field based technicians. Their movements are initiated by telephone calls from clients requiring service, followed by scheduling at a single, central point of service, and communication of the schedules to the technicians in the field by telephone from a central office. The technicians then record the equipment and its location at the site and provide other information for service reporting. As an added bonus, the technicians also note other information relating to the sites visited; qualitative information, such as type of pub and clientele, presence of a restaurant etc., which is all useful for formulating new marketing and business plans and planning new sales campaigns".

In this early example of a service management system (SMS), we can see the main functional elements that are evident in today's SMS packaged software modules. In essence these are:

- The scheduling of the delivery and/or installation of goods, services or contract work.
- The scheduling of human resources to undertake tasks related to those goods or services (installation, repair, maintenance, product upgrades, contract staff, project management)
- The processing of requests and inquiries from customers relating to the above mentioned goods and services.
- The two way exchange of marketing, sales and operational information between the customer and company personnel (located either on the customer site or at head office).
- Reporting of management and operational information on financial value of assets, call-out rates, problem resolution performance etc.

ISBN: 978-1-61208-152-6

It is these basic functions that constitute the core of today's SMS software, complemented by current data transfer and communications technology. It is worth making clear here that SMS as a concept in this paper is distinct from the concept of 'service' as used in the context of service-oriented architecture (SOA), which is a technology design concept concerned with the way in which new systems are developed and integrated. Brown, Johnston, and Kelly [5] have defined SOA as "a way of designing a software system to provide services to either end-user applications or other services through published and discoverable interfaces". There is, therefore, very little overlap with SMS as used in this paper, other than that SOA could be used as a design principle for developing SMS software.

The three SMEs studied in this research are located in the Midlands counties of Hereford and Gloucestershire. They are all of approximately the same size, having turnover of between £2.5 and £4.5m and staff numbers of 35 – 50. They are in three distinct industry sectors, all of which embody significant service management activities.

TPG DisableAids is a provider of equipment for the elderly and disabled and has grown steadily since its foundation in 1984 to employ 47 staff today. The company assembles and distributes a wide range of products from primary manufacturers, such as Stannah, who make a range of stair lift products. The company currently has an annual turnover of £4.3m (2009/10), with stair lift products generating about one-third of turnover but over 50% of profits.

TPG DisableAids' market can be divided into different segments (NHS, local authorities, district councils, residential & nursing homes, private individuals). Their business plan is to double their turnover within 5 years to £8.5m in 2014/15 which is dependent on the company having the systems capability to respond to the equipment and service requirements of the NHS and related bodies, as well as private individuals, at short notice as the elderly and disabled leave hospital and return to their homes. Service management processes and systems are key to achieving this turnover growth and increase in market share.

Optimum Consultancy Ltd was formed in 2008 through the merger of two companies - Hama Ltd, a project management services business, and J Orchard Consulting Ltd, a surveying services business. In its first trading year (2008-9) it achieved a turnover of £2.4m and this was increased to £3.1m in 2009-10. The company has 35 staff and its core business remains project and cost management in the property, engineering and construction fields; its customer base includes major retailers, rail operators, major financial and banking corporations and sustainable developments. The merger of the two companies posed significant challenges to combine and upgrade two different IT/IS architectures and in particular to align and standardise the customer facing processes and systems across three offices. In 2009-10, new integrated systems and customer facing processes were introduced to provide the infrastructure support needed for steady growth and improved margins, without the stop-start addition of administrative overheads. The systems strategy revolved around the Workspace integrated package (from Union Square software) and a major output was a new process and associated procedures for responding in a consistent and streamlined manner to customer enquiries, across the organization. This encompassed a review and evaluation of how Optimum's services and products could best be combined and supported to meet varying customer needs and improve customer service.

Muddy Boots Software Ltd (MBS) is a rapidly expanding software house and the company's business plan targets a trebling of turnover within 5 years from £1.6m in 2009 to £6.0m in 2014. The company has three main software products that are the main drivers of revenue growth in the immediate future (Quickfire, Greenlight and CropWalker). A strategic component of this growth is the implementation of customer centric systems and processes to drive and support new sales both in the UK and in overseas markets. A recent IS project has attempted to embed a new sales and marketing culture within the company based on the Microsoft CRM package and re-engineered supply chain and service management processes. MBS is moving from a mainly UK customer base to an international user base. This is supported by additional offices abroad and systems which can be accessible from multiple locations and time zones to enable a variety of services and support to be provided 'anytime, anywhere, any place'.

The paper is divided into five main sections. Following this introduction, section two reviews literature pertinent to the related themes of process change and alignment and the emergence of packaged software. The case study methodology is then briefly discussed in section three, followed by an overview of findings from the three case studies. Finally, in section five, some concluding remarks are made about the emergence of service management as a core business process and mainstream software package.

II. Literature Review

In this section, literature relating to the emergence of process thinking is discussed, leading to consideration of the parallel growth of business packaged software since the early 1990s. The concept of strategic alignment is then briefly examined in the context of process improvement and IS strategy development.

A. *Process versus Function*

Many companies want to organize themselves around processes but they do not have any clear idea which steps to follow and which initiatives need to be taken. Others are not sure how to structure a company around processes and sometimes turn to consultancy to help them decide what to do. There are also companies that are not sure whether their current organizational structure is adequate to engage process management. Earl [6] discusses how business processes can be improved by deploying new information systems. This paper focuses on in depth case studies at the three companies noted above, all of which have invested in information systems to support the service management process, in order to drive business expansion.

Companies sometimes find it problematic to adequately embrace process management especially when the organization is dominated by traditional functional structures. Organizations structured by task and function need to be redesigned to work by process. Some companies tend to take a few steps and give up without knowing how to progress. The difficulties stem from a poor understanding of the concept of process. Ramaswamy [7] suggests that companies that provide services normally think that process is a sequence of activities needed to perform transactions that help to provide their services.

To organize a company around business processes, it is necessary to focus on external customers because business processes usually start and end with them. Processes are a series of activities which begins with an exact understanding of what the external customer wishes and finishes with the external customer gaining what he/she needs and requests. The customer is always central within organizations structured by process and the final objective of these companies is to offer to the customer more value in less time and with less cost. Organizations are in a battle to achieve it and they are learning to think in new ways to structure the company accordingly.

To define processes is a difficult task, which involves many complex factors like customers, human behavior and company structure. However, process modeling can provide a less detailed way to define process. It has been suggested that "the task of modeling, in general, aims to provide, an abstract description of one slice of reality by omitting details and thus reducing complexity which is usually inherent in real world situations" [8]. In practice, functional areas do not disappear when companies organize themselves around processes. When process owners assume their responsibilities for specific projects, with related structure and process roles, the functional area bosses are left to focus on staff training and resource planning and management.

B. *Business Packaged Software*

Packaged software for most mainstream business processes came to market in the 1990s as the spread of the UNIX operating system as a *de facto* standard for mini computers and the increasing dominance of the Intel chipset led to a massive surge in the packaged software market. Building on the earlier Materials Requirements Planning (MRP) packages, other packaged software systems provided modules for sales order processing, ledgers, payroll and personnel as well as MRP, sometimes combined into one integrated package from one vendor – for example, the ERP software suites of Oracle and SAP. The increased take-up of packaged software coincided with the spread of business process re-engineering (BPR) as a management concept employed by many companies to improve efficiencies and reduce overheads. The two became closely linked as BPR projects were frequently combined with the introduction of new software solutions.

Only in recent years, however, has the concept of service management as a separate software module been clearly identified as a component of an integrated ERP solution or as a standalone solution as part of a 'best of breed' IS strategy. There is an on-going debate regarding whether 'best of breed' or 'one integrated package' (ERP) is the optimum solution for SMEs. Robinson [9] questions "if a company has, for instance, a financial package they are happy with, should they dump this and use the integrated package? The answer to this question is a definite 'yes'. If a fully integrated package is right for the company, the people using stand alone systems must be trained in the use of the integrated system so that all the stand alone systems, databases and spreadsheets can be dumped. Not only will islands of data reduce the advantages of the integrated system but it will also undermine its integrity".

C. *Strategic Alignment and Process Improvement*

Many of the theories and models of information systems strategy development are based on a logical progression from business strategy to evaluation of information requirements, leading to an information system (IS) strategy. Process analysis, producing 'current' and 'new' process maps, also features in some IS strategy development models; and data analysis and data modeling can also play an important role in determining what systems are required. Robson [10] suggests an information strategy is also appropriate.

Levy and Powell [11] have identified competitiveness and the importance of customer power in determining the way SMEs use IS, and they conclude that "a major barrier to the use of IS to support innovation is the leadership and technical knowledge of the owner and/or management team". Alignment of IS Strategy with overall business strategy is key. Koopman [12] supports this perspective in asserting that "the real threat to most companies is not a strategic threat from outside. Instead it is their own failure to align their organization with their strategy and thus ensure good execution". An organization must be able to align itself with the strategic plan and turn strategy into action.

As noted above, BPR is often regarded as a route to gaining the combined benefits of new information systems and process redesign and improvement. "BPR is the means by which an organization can achieve radical change in performance as measured by cost, cycle time, service, and quality, by the application of a variety of tools and techniques that focus on the business as a set of related customer-oriented core business processes rather than a set of organizational functions" [13]. Short and Venkatraman

[14] suggest that BPR has in the past largely had an internal, operational focus. The objective has usually been the optimization of a single process rather than transformation of the enterprise itself.

III. Research Questions And Methodology

Service management is a key activity in all three SMEs studied in this paper, even though it is recognized as a core process in only one of the three (TPG DisableAids). This paper aims to answer three research questions:

- What is the service management process in these three SMEs? Are there major differences or is the process essentially the same in these companies that operate in three different industry sectors?
- What information systems and support technologies are used to support the service management process?
- How effective are these systems and are they well matched to the specific requirements of service management in each company?

A. *Research Method*

Saunders, Lewis, and Thornhill [15] highlight that qualitative and inductive research can be done in different ways encompassing case studies, grounded theory, and ethnography. Remenyi, Williams, Money, and Swartz [16] agree that case studies are likely to be used as part of an inductive research approach. Cassell and Symon [17] define case study research as a "detailed investigation … of one or more organizations, or groups within organizations, with a view to providing an analysis of the context and processes involved in the phenomenon under study". The research method employed here is case study research of SMS systems implementations in three SMEs.

The evaluation of results was done comparing the answers given in questionnaires with observations made by the authors. This was analyzed in conjunction with the findings of a literature review, allowing empirical generalizations and a series of clear statements to be developed.

B. *Data Collection Methods*

There are several ways to collect qualitative data that have a case study research focus. Examples include questionnaires, interviews and observation. After carefully analyzing data collection methods, questionnaires and observation, combined with first hand interviews, were elected as the most suitable approaches for data collection due to the presence of the authors as consultants to the companies over a period of years.

Wu, Mahajan, and Balasubramanian [18] suggest that subjects must be in a position to generalize about business behavior. Only employees at the level of manager/supervisor or higher were considered for participation in the study. All responses were complemented by face to face interviews. In addition, at least one year of observations were also documented by the authors in each of the three companies, which helped the research by highlighting facts that were not gathered in the interviews.

TABLE 1. Service Management Activities At The Three Case Study Companies

Five main service management activities	Case Study Response
1.Scheduling of goods and services	TPG - YES OCL – YES MBS - YES
2. Scheduling of human resources	TPG - YES OCL – YES MBS - YES
3. Processing of customer requests, enquiries, and cost estimates	TPG - YES OCL – YES MBS - YES
4. Exchange of sales, marketing and operational information	TPG – LIMITED OCL – YES MBS - YES
5. Management and operational reporting	TPG - YES OCL – YES MBS - YES

IV. Findings

This section reports on the detailed findings from the questionnaire responses, personal observation and follow-up interviews in each of the three companies studied, looking at processes, systems and then functionality and integration issues.

A. *The Service Management Process in the Case Study Companies*

Questionnaire responses revealed interesting differences in how the three companies perceived service management. At TPG, service management was seen as one of the company's main business processes, closely linked to route planning and scheduling (for the company's 20 field based service engineers) and marketing and selling (Fig. 1). At OCL, however, service management was not seen as a distinct process, but rather as part of three other processes - business development, people management and operations management (Fig. 2). Similarly at MBS, service management is perceived as being a part of three main business processes – technical services, commercial services and sales and marketing (Fig. 3).

Nevertheless, all three companies are involved in the five main activities or functions that are generally covered by SMS and the constituent parts of the overall service management process (Table 1). Perhaps not surprisingly, given that the TPG management recognizes service management as a core process, the activity descriptions contained in the questionnaire response is somewhat richer than in the other two cases. For example, for the scheduling of goods and services, the respondent noted that "there are three administrators/clerks in the service department with roughly one third each of the geographic area we cover. Each has a team of engineers to manage and arrange

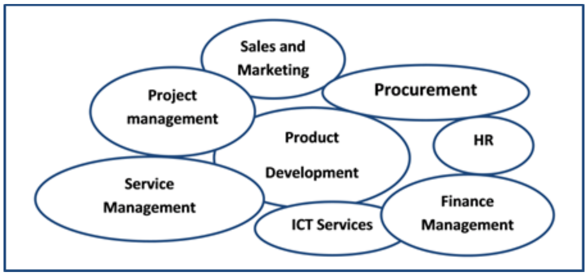


Figure 1. Business processes at TPG DisableAids.

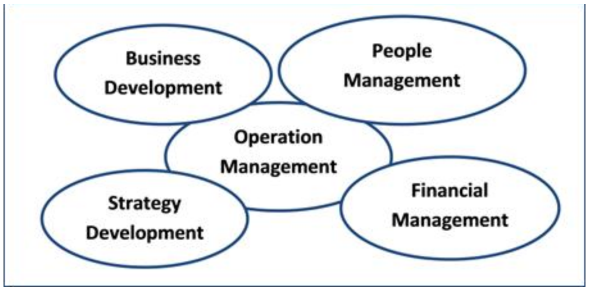


Figure 2: Business processes at Optimum Consultancy.

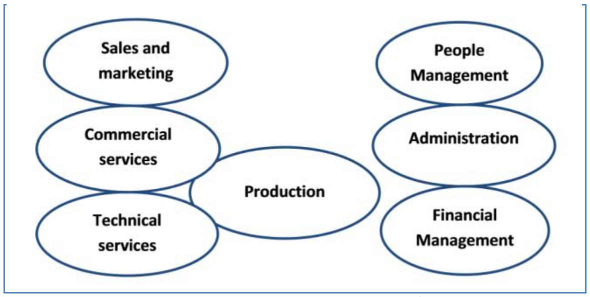


Figure 3: Business processes at Muddy Boots Software.

appointments for (a) scheduled installation of goods sold by the sales department (e.g., stair-lifts, hoists, etc.), (b) routine maintenance (e.g., 6 monthly service calls), (c) breakdowns (with response times being either same day, next day, 7 days, 14 days). Each month 'VSM' (the service management system) produces a list of all items that need to be serviced in that month. These are printed, split and allocated to each clerk. Appointments are made by agreement with either (a) a private customer on the phone, (b) with equipment owners where public sector organisations own the equipment. For breakdowns, each clerk may 'borrow' an engineer from another area” [19].

This activity description illustrates the type of specialist functionality that an SMS requires, combining call scheduling, stock management, field communications, field-based plant maintenance, customer management, and human resources scheduling. Indeed, as regards this last point, the respondent at TPG notes “delivery of goods not requiring expert engineering/installation personnel is undertaken by stores/warehouse…VSM contains all pending work and each clerk manages their own work in VSM and their own engineers” [19].

This contrasts with the process analysis at OCL where service management is not recognized as a core process. Nevertheless, the breakdown of main processes to activity level shows a range of functions that collectively may be seen, in part at least, as constituting the service management process – sales enquiry and tenders management, personnel management, project management, contacts management and marketing (Fig. 4). A similar picture emerges at MBS, where the service process activities are split across several main processes areas. Technical services and commercial services undertake activities such as the scheduling of human resources, the processing of customer requests, enquiries, and cost estimates, the scheduling of goods and services and the management and operational reporting. In addition the exchange of marketing and sales information is an activity carried out by the sales and marketing department and operational and management reporting also covers activities undertaken by the sales, marketing and finance departments.

B. *Service Management Systems Deployed in the Case Study Companies*

In all three companies, packaged software is deployed to support the main activities that make up the service management process (Table 2). At TPG, Vision Service Manager (VSM), from software house Sybiz, supports all these activities to some degree, although the perception is that this is only done moderately well – and that replacement will likely be required in the short to mid-term. This is partly because of the old FoxPro database technology that underpins the version of Sybiz used at TPG. The questionnaire respondent notes that “the entire software architecture and key processes are under review” [19].

At OCL, where the Workspace software package has recently been implemented, all main service management activities are seen as being well supported. Lau, Wynn, and Maryszczak [20] note that the system “allows the senior management team to track and manage the new work pipeline more easily. When Optimum wins a job, it is migrated to a project record form which holds all relevant history. Previously, as a job flowed through the Optimum business from enquiry to completion, it was recorded and tracked using spread sheets held on the network server. Staff spent a lot of time each month searching for or collating information relevant to a particular job. With the previous system, there was no way of sitting at one’s PC and getting a complete picture of any particular job across the various systems. A lot of manual processes were required to bind together project data from the various applications and, furthermore, there was a lack of version control on these documents.” This is interesting because Workspace is generally positioned as being in the ‘collaboration management’ software niche rather than a ‘service management’ package.

At MBS, the picture is less clear. The Technical Services Manager at MBS noted that Microsoft CRM was the key

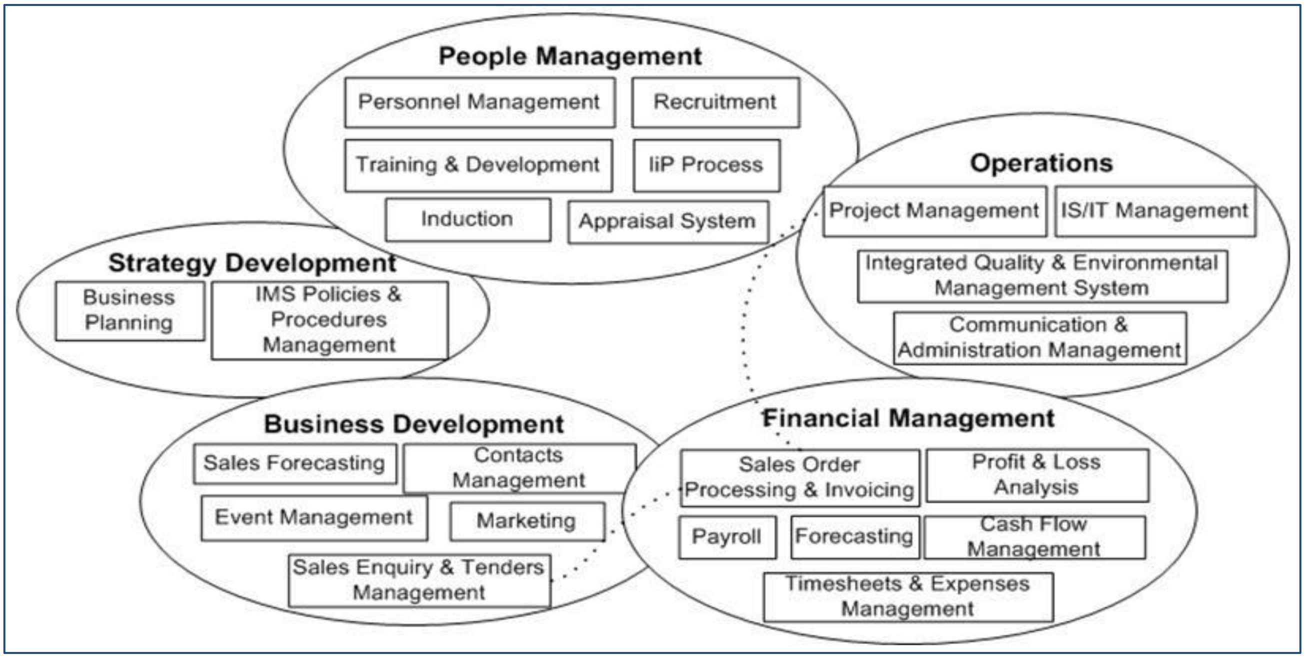


Figure 4 OCL processes broken down to activity level.

service management software in the company and is used alongside Microsoft Outlook with which it integrates. In addition, other departments that are involved in service management are using other packages (Microsoft TFS and a bespoke legacy system for timesheets), albeit in a lesser way to support this, with obvious disadvantages (no common point for service management or one-business view for reporting). "Microsoft CRM and Microsoft TFS have only recently been brought into use in this business and even though their capabilities are vast, they are not being fully utilised currently. Because service management is dispersed across several departments within the company and several systems are currently in use there are improvements to be made by integrating these systems more closely. Further work needs to be done to achieve this before the company grows further" [21].

It was felt that this situation was partly due to the introduction of new systems which take time to bed in. Thus a legacy system (such as the bespoke timesheet system) will be decommissioned when the new systems - Microsoft TFS and CRM - take over this functionality. When that occurs, these new systems can be integrated to give a one-business view of service management. The reason for multiple system use in service management at MBS can be related to the fact that this company does not view service management as a key standalone process. Consequently it is spread over several functional departments, each of which also has other activities to perform, and therefore departments have opted for systems which support their overall role rather than just service management. Overarching this is a top-down drive for business wide systems that will draw the constituent parts together through integration.

C. *Functionality and Integration Issues*

Table 2 reveals a mix of 'green' and 'amber' ratings for service management systems in the three companies. Respondents were asked to assess two main aspects – functionality (against requirements) and technical and integration issues. Understandably, it is at OCL that management are most satisfied as they have just recently implemented the integrated Workspace solution which covers the company's main service management requirements. At TPG, there are problems with all service management functions. As regards management information, the IT manager notes on the questionnaire that "report facilities within VSM are very poor. They are inaccurate and since the product does not have basic validation of data entered, most reports generate garbage. Additional software has had to be written to correct the data and report on that" [19].

However, he also highlights the benefits of integration: "Vision Service Manager (VSM) and Vision are written by the same supplier. The two products are 98% reliable in terms of integration but errors do occur and take a lot of time to correct". In conclusion, he asserts that "the entire software architecture and key processes are under review. Current systems do not cover all requirements" [19].

At MBS, as noted above, a number of different software products are used for service management activities due to service management being split across the main major business processes and therefore different activities are undertaken by different teams of staff. The majority of these are Microsoft products (TFS, Outlook and CRM) and it is thus possible to integrate them given enough resource. One additional package, a bespoke timesheet package, may not

TABLE 2. SYSTEMS DEPLOYED FOR SERVICE MANAGEMENT AT THE THREE CASE STUDY COMPANIES TABLE TYPE STYLES

Activity	Software package	Function ality match (R A G)	Technical/ Integration suitability (R A G)
The scheduling of the delivery, installation or maintenance of equipment, goods or contract work.	TPG - VSM OCL – Workspace MBS - Outlook, MS CRM, MS TFS	Amber Green Amber	Amber Green Amber
The scheduling of human resources to undertake service oriented tasks, relating to the installation, maintenance or delivery of products and services	TPG – VSM OCL – Workspace MBS - Outlook and MS TFS	Amber Green Amber	Amber Green Amber
The processing of requests and enquiries from customers relating to the above mentioned goods and services (including cost estimates, problem logging and resolution, call out requests)	TPG – VSM OCL – Workspace MBS – MS CRM and Outlook	Amber Green Green	Amber Green Amber
The two way exchange of marketing, sales and operational information between the customer and company personnel (either on-site or at head office).	TPG – VSM OCL – Workspace MBS – MS CRM and Outlook	Amber Green Green	Amber Green Amber
Reporting of management and operational information on financial value of assets, call-out rates, problem resolution performance etc.	TPG - VSM OCL – Workspace MBS -MS CRM, MS TFS, Bespoke timesheets	Amber Green Amber	Amber Green Amber

integrate but is under review anyway as it is felt that this functionality can probably be gained through Microsoft CRM and TFS. The impression at MBS is that service management is not currently being supported as well as it could be by existing systems because of the absence of a holistic view and comprehensive analysis of the requirements of the service management process. This means that service management is reported on in components across varying departments. There is a lot of technically sound software in place, but it now needs to 'bed in', and this may require additional analysis and a degree of re-implementation and will certainly require further development of the current systems for which there is the expertise within the company.

V. CONCLUDING REMARKS

A major software provider [22] asserts that "despite the increasing importance of service in driving business success, many companies operate with disconnected service processes, disparate data, and isolated point solutions, which result in inefficient, manual service processes, high maintenance costs, ineffective planning, and, eventually, growing customer attrition and lost revenue". The case studies assessed in this paper suggest that 'service management' is neither clearly defined nor universally recognized in the context of SME business operations, either as a process or as a software system. This is no doubt partly because of the plethora of new management and software concepts that have appeared in the past decade – customer relationship management, knowledge management, content management, collaboration management, workflow, supply chain management, demand chain management; there are only so many concepts that a management team can grapple with, particularly in SMEs where business success often derives from a hard-nosed, 'seat of the pants' approach to management.

Nevertheless, the main activities that we suggest make up the service management process *are* generally recognized at an operational level in all three case study companies, and this suggests that service management will increasingly be seen as a core business process. This will be engendered by the continued development and increased deployment of service management systems. At present, however, the matching of service management activity to software system is fragmented and not well co-ordinated, with resultant problems in customer servicing. It has been suggested that the "strategic alignment of IT exists when a business organisation's goals, activities and processes are in harmony with the information systems that support them" [23]. However, only at OCL is this arguably the current position; at TPG DisableAids, where they *do* have a service management process and system, additional technology is used to try to improve software performance, notably in the provision of management information, and a major IS strategy overhaul is likely soon.

Market pressures are forcing companies to improve their efficiency and in particular the effectiveness to deliver customer satisfaction. Given the strong customer centric dimension to service management, it seems probable that – both as a process and business software package - the concept will grow in significance in the management and operation of SMEs.

ACKNOWLEDGMENTS

Grateful acknowledgement is given to Phillip Turner, IT Manager at TPG DisableAids Ltd, and to Erin Lau, IS Projects Manager at Optimum Consultancy Limited, for providing information on service management in their respective companies. The university-industry partnerships at all three companies reported on in this article received financial support from the Knowledge Transfer Partnerships programme (KTP). KTP aims to help businesses to improve their competitiveness and productivity through the better use of knowledge, technology and skills that reside within the UK Knowledge Base. KTP is funded by the Technology Strategy Board along with the other government funding organisations.

REFERENCES

[1] J. Hurwitz, R. Bloor, M. Kaufman, and F. Halper, Service Management for Dummies, Hurwitz Associates, 2009.

[2] Www.webopedia.com/.../strategic_service_management Html, accessed July 4th, 2011.

[3] Www. Solavista.com, accessed July 4th, 2011.

[4] R. Bloor, Corporate Computer Strategy, Butler Bloor Ltd, 1993, pp159-160.

[5] A. Brown, S. Johnston, and K. Kelly, Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications, Rational Software Corporation, 2002.

[6] M. J. Earl, Information Management: The Strategic Dimension, Oxford University Press, Oxford, 1988.

[7] R. Ramaswamy, Design and management of service processes, Addison Wesley, Reading, 1996.

[8] A. Tsalgatidou and S. Junginger, "Modeling in the Re-engineering Process," ACM SIGOIS Bulletin, 1995, pp. 17-24.

[9] P. Robinson, "Best of Breed (BOB) v Fully Integrated (FIS)," BPIC consultancy, Hove, 2003; accessed on internet at URL: http://www.bpic.co.uk April 7th, 2011.

[10] W. Robson, Strategic Management of Information Systems, Pearson Education, Harlow, 1997.

[11] M. Levy and P. Powell, "SME Internet Adoption: Towards a Transporter Model," International Journal of Electronic Commerce and Business Media, 13, 2003, pp. 173-181.

[12] J.C. Koopman, "Effective alignment: strategy cannot succeed without it," Canadian Manager, 1999, pp. 14-5.

[13] H. Johansson, P. McHugh, A. Pendlebury, and W. Wheeler, Business Process Reengineering – Breakpoint Strategies for Market Dominance, Wiley, Chichester, 1993.

[14] J. Short and N. Venkatraman, "Beyond business process redesign: redefining Baxter's business network," Sloan Management Review, 1992, pp. 7-21.

[15] M. Saunders, P. Lewis, and A. Thornhill, Research methods for business students, Prentice Hall, Harlow, 2003.

[16] D. Remenyi, B. Williams, A. Money, and E. Swartz, Doing research in business and management, an introduction to process and method, Sage Publications, London, 1998.

[17] C. Cassell and G. Symon, Qualitative Methods in Organizational Research: A Practical Guide, Sage Publications, London, 1994.

[18] F. Wu, V. Mahajan, and S. Balasubramanian, "An Analysis of E-business Adoption and its impact on business performance," Journal of the Academy of Marketing Science, volume 31, no 4, 2003, pp. 425-447.

[19] P. Turner, Questionnaire on the service management processes, related activities and systems support at TPG DisableAids, May 10th, 2011 (unpublished).

[20] E. Lau, M. Wynn, and P. Maryszczak, "Enterprise application integration in a service industry SME: a case study of Optimum Consultancy Services," in Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference, IEEE Explore, pp. 71-76, ISBN: 978-1-4244-8068-5.

[21] E. Tipton, Questionnaire on the service management processes, related activities and systems support at Muddy Boots Software, May 13th, 2011 (unpublished).

[22] SAP, Business benefits of SAP service management, www.onestopsap.com/SAP-SM, accessed July 6th, 2011

[23] S. Bleistein, K. Cox, J. Verner, and K. Phalp, "B-SCP: a requirements analysis framework for validating strategic alignment of organisational IT based on strategy, context and process," Information and Software Technology, 2006, Elsevier, pp. 846-868.