IARIA

SERVICE COMPUTATION 2013

The Fifth International Conferences on Advanced
Service Computing

ISBN: 978-1-61208-270-7

May 27- June 1, 2013

Valencia, Spain

SERVICE COMPUTATION 2013 Editors

Arne Koschel, Fachhochschule Hannover, Germany

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

SERVICE COMPUTATION 2013

Foreword

The Fifth International Conferences on Advanced Service Computing (SERVICE COMPUTATION 2013), held between May 27 and June 1, 2013 in Valencia, Spain, 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 and integration. Some services might require energy optimization, some might require special QoS guarantee in a Web-environment, while others might need 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 2013 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 2013. 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 2013 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 2013 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 advanced service computing.

We are convinced that the participants found the event useful and communications very open. We hope that Valencia, Spain provided a pleasant environment during the conference and everyone saved some time to explore this historic city.

SERVICE COMPUTATION 2013 Chairs:

SERVICE COMPUTATION General Chair

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

SERVICE COMPUTATION Advisory Chairs

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 2013 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 2013

Committee

SERVICE COMPUTATION General Chair

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

SERVICE COMPUTATION Advisory Chairs

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 2013 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 2013 Technical Program Committee

Witold Abramowicz ,Poznan University of Economics, Poland
Dimosthenis S. Anagnostopoulos, Harokopio University of Athens, Greece
Julian Andrés Zúñiga, Ingeniero en Electrónica y Telecomunicaciones / Unicauca, Colombia
Ismailcem Budak Arpinar, University of Georgia, USA
Johnnes Arreymbi, School of Architecture, Computing and Engineering - University of East London, UK
Irina Astrova, Tallinn University of Technology, Estonia
Jocelyn Aubert, Public Research Centre Henri Tudor, Luxembourg
Benjamin Aziz, School of Computing - University of Portsmouth, UK
Youcef Baghdadi, Department of Computer Science - Sultan Qaboos University, Oman
Zubair Ahmed Baig, KFUPM - Dhahran, Saudi Arabia
Gabriele Bavota, University of Sannio, Italy
Ali Beklen, IBM Turkey - Software Group, Turkey
Oualid Ben Ali, University of Sharjah, UAE
Emmanuel Bertin, Orange-ftgroup, France
Sergey Boldyrev, Nokia, Finland
Juan Boubeta Puig, University of Cádiz, Spain
Massimo Cafaro, University of Salento, Italy
Radu Calinescu, Aston University-Birmingham, UK
Chin-Chen Chang, Feng Chia University, Taiwan

Maiga Chang, Athabasca University, Canada
Rong N. Chang, IBM T.J. Watson Research Center, U.S.A.
William Cheng-Chung Chu, Tunghai University, Taiwan
Giuseppe De Pietro, Institute for High Performance Computing (ICAR) / National Research Council of Italy (CNR) - Napoli, Italy Manuel
Andrea Delgado, University of the Republica, Uruguay
Leandro Dias da Silva, Federal University of Alagoas, Brazil
Kamil Dimililer, Near East University, Cyprus
Erdogan Dogdu, TOBB University of Economics and Technology - Ankara, Turkey
Juan Carlos Dueñas López, Universidad Politécnica de Madrid, Spain
Haikal El Abed, Technische Universitaet Braunschweig, Germany
Vincent C. Emeakaroha, University College Cork, Ireland
Onyeka Ezenwoye, Georgia Regents University, USA
Massimo Ficco, Second University of Naples, Italy
Sew Bun Foong, National University of Singapore, Singapore
Steffen Fries, Siemens Corporate Technology - Munich,, Germany
G. R. Gangadharan, Institute for Development & Research in Banking Technology [IDRBT] - Hyderabad, India
Parisa Ghodous, Claude Bernard University of Lyon, France
Joseph Giampapa, Carnegie Mellon University’s Software Engineering Institute USA
Wasif Gilani, SAP Research, UK
Luis Gomes, Universidade Nova de Lisboa / UNINOVA-CTS - Monte de Caparica, Portugal
Andrzej Goscinski, Deakin University, Australia
Gustavo González, Mediapro Research - Barcelona, Spain
Andrzej M. Goscinski, Deakin University - Victoria, Australia
Victor Govindaswamy, Texas A&M University-Texarkana, USA
Mohamed Graiet, Institut Supérieur d'Informatique et de Mathématique de Monastir, Tunisie
Maki K. Habib, American University in Cairo, Egypt
Ileana Hamburg, IAT - Westfälische Hochschule Gelsenkirchen, Germany
Sven Hartmann, Clausthal University of Technology, Germany
Martin Henkel, Department of Computer and Systems Sciences – Stockholm University, Sweden
Bernhard Hollunder, Hochschule Furtwangen University - Furtwangen, Germany
Wladyslaw Homenda, Warsaw University of Technology, Poland
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Wei-Chiang Hong, Zhejiang University of Finance & Economics, China
Sun-Yuan Hsieh, National Cheng Kung University, Taiwan
Paul Humphreys, Ulster Business School/University of Ulster, UK
Mirjana Ivanovic, University of Novi Sad, Serbia
Hemant Jain, University of Wisconsin- Milwaukee, USA
Jinlei Jiang, Tsinghua University - Beijing, China
Ivan Jelinek, Faculty of Electrical Engineering - Czech Technical University Department of Computer Science and Engineering, Czech Republic
Alexander Jungmann, University of Paderborn, Germany
Rajaraman Kanagasabai, Institute for Infocomm Research, Singapore
Tahar Kechadi, University College Dublin, Ireland
Nhien An Le Khac, University College Dublin, Ireland
Hyunsung Kim, Kyungil University, Korea
Alexander Kipp, Robert Bosch GmbH, Germany

Manuele Kirsch Pinheiro, Université Paris 1 - Panthéon Sorbonne, France
Mourad Kmimech, l'Institut Supérieur d'informatique de Mahdia (ISIMA), Tunisia
Arne Koschel, University of Applied Sciences and Arts - Hannover, Germany
Yousri Kouki, ASCOLA - INRIA, France
Natalia Kryvinska, University of Vienna, Austria
Patricia Lago, VU University Amsterdam, Netherlands
Ulrich Lampe, Technische Universität Darmstadt, Germany
Annett Laube-Rosenpflanzer, Bern University of Applied Sciences - Biel/Bienne, Switzerland
Guanling Lee, National Dong Hwa University, Taiwan
Keqin Li, SAP Research, France
Kuan-Ching Li, Providence University, Taiwan
Noura Limam, University of Waterloo, Canada
Qing Liu, The Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia
Shih-His (Alex) Liu, California State University - Fresno, USA
Hui Ma, Victoria University of Wellington, New Zealand
Sabri A. Mahmoud, King Fahd University of Petroleum and Minerals, Saudi Arabia
Kurt Maly, Old Dominion University, USA
Gregorio Martinez, University of Murcia, Spain
Lena Mashayekhy, Wayne State University, USA
Mihhail Matskin, KTH, Sweden
Manuel Mazzara, UNU-IIST, Macao & Newcastle University, UK
Viktor Medvedev, Vilnius University, Lithuania
Souham Meshoul, Constantine Mentouri University, Saudi Arabia
Fernando Moreira, Universidade Portucalense, Portugal
Haris Mouratidis, University of East London, UK
Debajyoti Mukhopadhyay, Maharashtra Institute of Technology, India
Francisco Javier Nieto De-Santos, Atos Research and Innovation - Bilbao Spain
Mara Nikolaidou, Harokopio University of Athens, Greece
Roy Oberhauser, Aalen University, Germany
Matthias Olzmann, noventum consulting GmbH - Münster, Germany
Hichem Omrani, CEPS/INSTEAD - GEODE dept., Luxembourg
Claus Pahl, Dublin City University, Ireland
Ingo Pansa, iC Consult, Germany
Namje Park, Jeju National University, Korea
Dana Petcu, West University of Timisoara, Romania
J Brian Pickering, IT Innovation Centre, UK
Pasqualina Potena, Università degli Studi di Bergamo, Italy
Thomas E. Potok, Oak Ridge National Laboratory, USA
Thurasamy Ramayah, Universiti Sains Malaysia, Malaysia
Juan J. Ramos-Munoz, University of Granada, Spain
José Raúl Romero, University of Córdoba, Spain
Feliz Ribeiro Gouveia, Fernando Pessoa University, Portugal
Juha Röning, University of Oulu, Finland
Gustavo Rossi, Universidad Nacional de La Plata, Argentina
Michele Ruta, Politecnico di Bari, Italy
Dieter Schuller, Technische Universität Darmstadt, Germany
Frank Schulz, SAP Research Karlsruhe, Germany
Xu Shao, Institute for Infocomm Research, Singapore

Copyright Information

Table of Contents

Structure

Lianyong Qi, Xiaona Xia, Jiancheng Ni, Chunmei Ma, and Yanxue Luo

myIdP - The Personal Attribute Hub

Annett Laube and Severin Hauser
Bern University of Applied Sciences
Devision of Computer Science
Biel/Bienne, Switzerland
Email: annett.laube@bfh.ch, severin.hauser@bfh.ch

***Abstract*—The myIdP service is an extension to the Swiss eID infrastructure with the aim to provide a service that handles personal attributes (like address, telephone number, email), which are neither part of the SuisseID identity providers nor of a Claim Assertion Services (CAS) because there is no official authority owning and certifying these data. The myIdP service is a (pseudo-)local CAS that can reuse data, which a user has already given to an application via an Internet transaction. The data is thus validated by the web application before being transferred - as SAML 2.0 attribute assertion - to the myIdP service. The myIdP service comes in two flavors with different trust relations: the attribute provider and the claim proxy. The attribute provider unites several claims for a given attribute and provides an optional quality assessment before sending it to a requesting web application. A trust relationship must consist between myIdP and the web application. The claim proxy only collects the received claims for a given attribute and transfers them with all details to the requesting application. The application can evaluate the confidence in the data based on the claim details. The myIdP service is evaluated in a scenario of prefilling e-forms in a eGovernment application.**

Keywords*—*electronic identity; SuisseID; attribute authority; e-form

I. Introduction

Like in many European countries, also in Switzerland an infrastructure for electronic proof of identities (eID) was developed and introduced in 2010 as SuisseID Infrastructure. The basis is the SuisseID [1] available as USB stick or chip card containing two digital certificates: (1) the SuisseID identification and authentication certificate (IAC) and (2) the SuisseID qualified digital certificate (QC). The SuisseID IAC can be used to identify the owner in Internet transactions. The SuisseID QC can be used to sign electronic documents in a forgery-proof manner. The SuisseIDs are issued by identity providers (IdP). In contrary to other European countries, where electronic identities are issued by the government together with offline identification (ID card), there are actual three commercial and one governmental SuisseID IdP.

The SuisseID itself and its certificates contain only a minimum of personal data (SuisseID number, name or pseudonym and optional email address) due to stringent privacy and data protection requirements in Switzerland. Additionally, a subset of the personal data from the identification document (e.g., a passport) and a well-known set of additional attributes gathered during the registration process (so called Registration process data, RPD) are stored in the identity provider service (extended IdP). The only way to retrieve this data is by strong authentication with the IdP service using the appropriate SuisseID IAC. The SuisseID Infrastructure is completed with a set of Claim Assertion Services (CAS) [2]. The purpose of a CAS is to provide and certify specific properties or attributes, which had been assigned to the SuisseID owner by some private or public authority. Examples are the membership of an organization or a company, and the proof of professional qualifications, like a notary or a doctor. Especially in the context of eGovernment, there is a need for an extension of the beforehand described SuisseID Infrastructure.

Many more personal attributes (like invoice address, telephone number, email) used in web applications, e.g., online shops, or in electronic forms often used in the eGovernment, are neither subject of the SuisseID IdPs nor the CAS because there is no official authority owning and certifying these data. The myIdP service fills that gap and allows storing and maintaining personal attributes for a SuisseID owner. The idea is to store information, which was at least entered (and thus used) once in a web application, for later reuse. The data is used and thus validated by the web application before being transferred as SAML attribute statement [3] (the so-called attribute claim) to the myIdP service. From now on, the user can reuse the attribute for other applications, which improves usability and reduces the error potential in the daily internet transactions.

The paper starts with the related work in Section II, before outlining the architecture and flavors of myIdP in Section III. In Section V, the integration of myIdP in a scenario of prefilling e-forms is shown. Section VI concludes the document.

II. Related Work

A service like myIdP or a SuisseID CAS corresponds technically to an Attribute Authority defined by SAML [3]: An Attribute Authority is a system entity that produces attribute assertions.[4]

In general, most of the known SAML Identity Providers (IdPs) can act as an Attribute Authority and issue attribute assertions beside their usual authentication functionality. Examples are the government-issued electronic identities of the European Countries, like the German Identity Card [5], the beID from Belgium [6] or the Citizens Card from Austria [7]. Similar to the SuisseID, all these government-issued eIDs provide only a small number of personal attributes related to the identity document they belong to.

The national electronic identities of the European member states are made interoperable with the STORK European eID Interoperability Platform [8]. With six pilots, the STORK project offers several cross-border eGovernment identity services. In the follow-up project STORK2.0 [9] also personal attributes related to eIDs are subject of investigation. E.g., in the banking pilot, public and private identity and attribute providers are included in the process of ”Opening a bank account” in a foreign country online with a national eID without physical presence. myIdP could be used in this context as attribute provider for personal attributes, like address, telephone number, email, etc.

In contrast to the central, government-regulated eID services, OpenID [10] is a decentral authentification service for web based services. The user is free to choose his favorite OpenID identity provider to get a OpenID, which is an URL or XRI including an end-user chosen name (e.g., alice.openid.example.org). OpenID providers are, e.g., Clavid [11] , CloudID [12], Google [13] etc. The OpenID providers themselves can support different authentification methods. For example, Clavid offers username/password, one time passwords, SuisseID authentication and the biometric AXSionics Internet Passport.

User attributes, like name, gender or favorite movies, can be also transferred from the OpenID identity providers to the relying party following the OpenID Attribute Exchange Specification [14]. The attributes can be (almost) freely defined according to the requirements of the relying party. As many OpenID providers do not validate the information entered by the users, the provided attributes have a low level of assurance. There is a need for a validation by a trusted 3rd party, a so called attribute provider (AP). Google started the Open Attribute Exchange Network (Open AXN, also known as ”street identity”, see [15]) to include validated information from APs. myIdP has the potential to act as an OpenID attribute provider, but is currently only enabled for use together with the SuisseID.

WebIDs [16] are especially common in social media (FaceBook, LinkedIn . . .) to allow users to identify themselves in order to publish information. Each user can make his own WebID or rely on an identity provider. The WebID is a URL with a #tag pointing to a foaf file [17] containing a cross-link to a (self-generated) certificate. Information that should be included but are not required to be present in a WebID Profile are the name (foaf:name) of the individual or agent, the email address associated (foaf:mbox) and the agent’s image (foaf:depiction). More attributes and links to all kind of web objects (other persons, groups, publications, account, . . .) can be also included. WebIDs can be connected to OpenIDs and vice versa.

III. Architecture

myIdP consists of four components (see orange boxes in Figure 1): the myIdP Service, the myIdP WebApp, the myIdP Admin and the myIdP API.

The **myIdP Service** is an attribute authority according to the SAML 2.0 [3] standard distributing assertions in response to identity attribute queries from an entity acting as an attribute requester. Like a typical SuisseID CAS [2], the myIdP service let the users select and confirm the properties, which were formerly received from an attribute issuer and are stored in the myIdP database.

New, to the concept of CAS, is the provisioning of a quality (level of assurance, level of confidence) together within the attribute assertions. myIdP integrates a quality module, that calculates the trustworthiness of the provided information on the basis of the ages, number of affirmations and quality of the issuer of the received and stored attribute assertions. This assurance level or quality can be used by an attribute requester to insist on a certain level of assurance for the requested attributes. The calculation of the assurance level or quality is a research topic on its own and therefore not included in this paper. A possible approach is shown in [18].

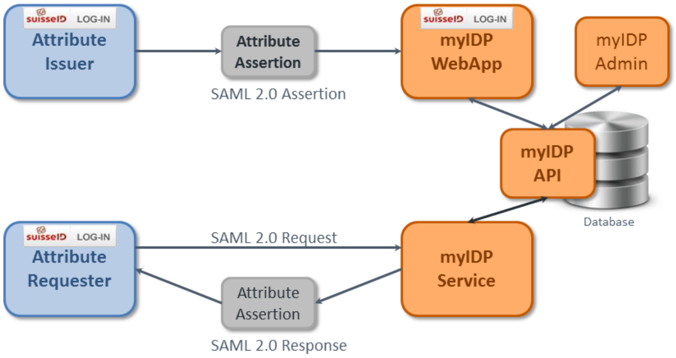


Figure 1. myIdP components and service provider roles

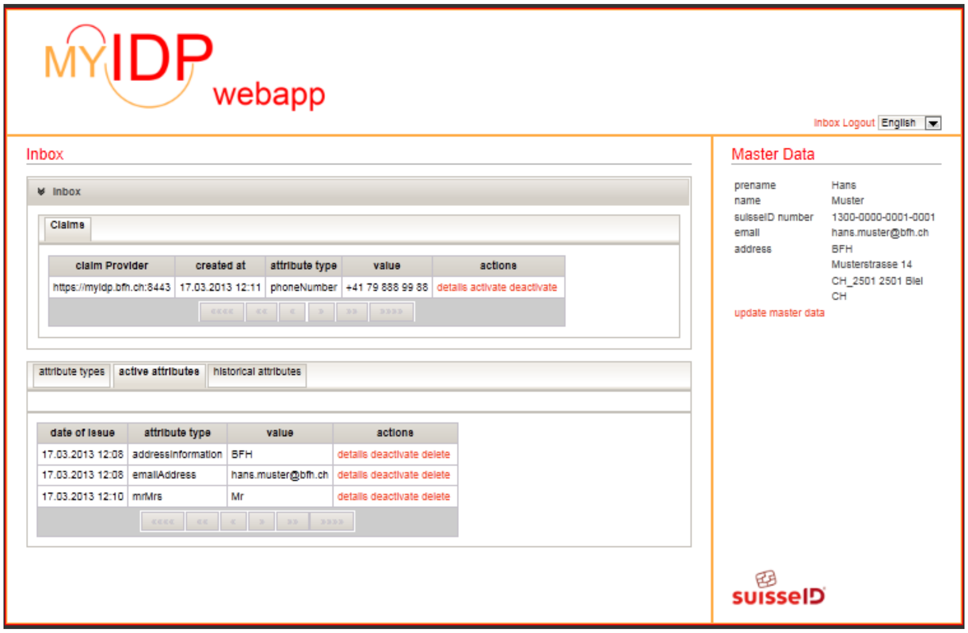
The **myIdP WebApp** is the end user front-end of myIdP where users – after the successful authentication with their SuisseID IAC – can view and manage their attributes. Attributes can not be entered directly in the myIdP WebApp, except the master data related to the myIdP account. Attributes always come from a service provider, e.g., a web application, acting as attribute issuer, which forwards – after confirmation by the user – the attribute assertions to myIdP. The attributes then arrive in the so-called **Inbox** (see Figure 2) where they can be detailed viewed and manually activated before they are exposed via the myIdP service. Corresponding to the user centric approach of the SuisseID, the users are always in control of their data and can activate/deactivate and delete attributes at any time. As a side effect, the user gets an usage history of his attributes in the web.

The **myIdP Admin** is an administration tool for myIdP. It supports the maintenance of attribute definitions and the registration process of service providers, which is needed to set up secure connections and trust relationships. New attributes can be enabled for usages simply by importing the related XML Schemata or by the use of the SAML Metadata Exchange [19].

The **myIdP API** provides an interface to the central database used commonly by the other three myIdP components.

A service provider can interact with myIdP incorporating two roles (see the blue boxes in Figure 1):

- **Attribute Requester:** The service provider electronically sends an attribute query to the myIdP service in order to draw a confirmation statement - a SAML 2.0 attribute assertion - from the myIdP service and uses it in further actions, e.g., prefilling of web forms.
- **Attribute Issuer:** The service provider sends SAML



claim Provider	created at	attribute type	value	actions
https://myidp.bfh.ch:8443	17.03.2013 12:11	phoneNumber	+41 79 888 99 88	details activate deactivate

date of issue	attribute type	value	actions
17.03.2013 12:08	addressInformation	BFH	details deactivate delete
17.03.2013 12:08	emailAddress	hans.muster@bfh.ch	details deactivate delete
17.03.2013 12:10	mrMrs	Mr	details deactivate delete

Master Data

prename	Hans
name	Muster
suisseID number	1300-0000-0001-0001
email	hans.muster@bfh.ch
address	BFH Musterstrasse 14 CH_2501 2501 Biel CH

update master data

Figure 2. Screenshot myIdP WebApp - Inbox

2.0 attribute statements to myIdP. (Despite the possibility to group several attributes in one SAML statement, myIdP prefers single attribute statements, in order to expose a minimum of information in the claim proxy case.) The attribute values where entered either manually by the users or requested beforehand from the myIdP service.
A special attribute provider is the myIdP WebApp, which uses the master data (address, email) entered during the myIdP registration process, to provide the first attribute statements to the users.

The myIdP service is available in two flavors: the **Attribute Provider** and the **Claim Proxy**.

The attribute provider summarizes the in the myIdP database available attribute assertions for the given request. All details about the original attribute provider of the information are hidden. After the user has selected and confirmed the attribute values, the newly built attribute assertion is signed by the myIdP service. When requested, an assurance level is included in this assertion. For this myIdP flavor, a direct trust relationship is established between the myIdP service and the web application in the attribute requester role.

This is different in the second myIdP flavor. The claim proxy extracts the stored attribute assertions from the myIdP database for a given attribute request. After the selection of attribute values and the explicit confirmation by the user, an attribute assertion containing an URI and optional the assurance level is returned to the requesting web application. This attribute request is also signed by myIdP but only to ensure integrity. The web application can use the URI from the attribute assertion to assess the originally received attribute assertions enveloped in an XML document. The web application can now access all details of the original assertions, including the issuers and timestamps, and perform its own quality assessment. The trust relationship has changed: the web application trusts directly the attribute issuers.

```
<complexType name="AttributeType">
  <sequence>
    <element ref="saml:AttributeValue"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Name" type="string"
    use="required" />
  <attribute name="NameFormat" type="anyURI"
    use="optional" />
  <attribute name="FriendlyName" type="string"
    use="optional" />
  <attribute name="myidp:Quality" type="decimal"
    use="optional" />
  <attribute name="myidp:ClaimListURI" type="anyURI"
    use="optional" />
  <attribute name="myidp:ClaimList" type="boolean"
    use="optional" />
  <anyAttribute namespace="##other"
    processContents="lax" />
</complexType>
```

Figure 3. Extended xsd `AttributeType`

In order to support the provision of a quality assessment of an attribute value and of the claim list URI, the SAML attribute assertions was extended (see the XSD fragment shown in Figure 3).

IV. PRIVACY

One important characteristic of myIdP (valid for both flavors) is the user-centric approach. The user is always aware which information is exchanged and has explicitly to confirm every single attribute, which is sent out by myIdP.

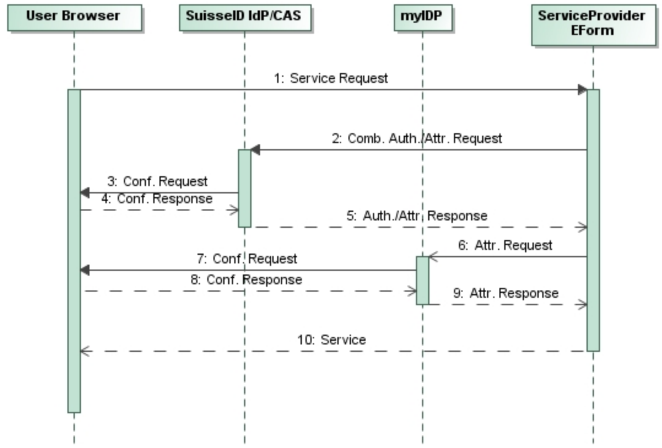


Figure 4. Sequence diagram "Get e-form"

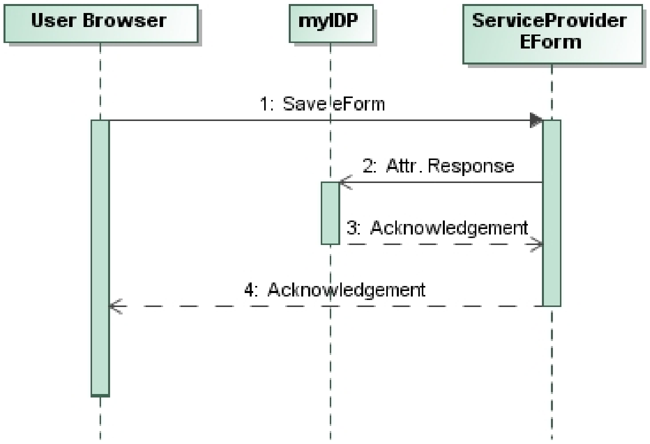


Figure 5. Sequence diagram "Save e-form"

myIdP implements multiple measures to ensure the privacy of the user: First, every Attribute Issuer has to get an user consent before sending any attribute statement to myIdP. Secondly, in myIdP the arriving attributes are deactivated by default. The user has explicitly to confirm if he wants to activate these attributes for further use. At any time, the user is free to delete attribute statements in the myIdP WebApp or to deactivate them. Thirdly, the user is involved in every message exchange with an Attribute Requester and has to confirm all attribute values. In the claim proxy case, he has also to confirm the disclosure of original attribute assertions. The attribute assertions contain information about visited web sites and could be used to track the user and to create user profiles. In any case, myIdP only sends attribute statements to Attribute Requesters only if there is a valid authentication with a SuisseID in addition to the user consent. These procedures ensures that the user exposes only the data he wants to use and protects his privacys.

V. APPLICATION SCENARIO

A scenario of completing electronic forms (e-forms) validates our approach. E-forms are commonly used in the Swiss eGovernment. With the help of the SuisseID, the citizen can be securely identified and the attributes stored in the core SuisseID components, like name, birthday, place of birth or nationality, can be used to prefill the e-forms. As the number of attributes available in the core SuisseID is quite limited, we want to use myIdP to provide additional values for the e-forms.

For our proof of concept, we chose the form "Proof of residence", which had already an integration with the core SuisseID infrastructure. In Figure 4, the interactions between the user, the e-form provider, the core SuisseID components and myIdP are depicted:

1) Service Request: the user requests an e-form from the e-form provider (e.g., by clicking on a link).
2) Authentication with SuisseID: the e-form provider issues an authentication and attribute request to the SuisseID IdP/CAS service. The following attributes are requested: name, first name and birthday.
3) Confirmation request: the user has to identify himself by entering his secret key (PIN) and in a second step to confirm his SuisseID attributes.
4) Confirmation response: the user's decisions are sent back to the SuisseID IdP/CAS.
5) Authentication and Attribute response: the SuisseID IdP/CAS sends a combined authentication and attribute assertion back to the e-form provider.
6) myIdP attribute request: the e-form provider issues an attribute request to the myIdP service asking for the address and the email.
7) Confirmation request: the user has to select the attribute values in case several emails or addresses are stored in myIdP and to confirm the selection.
8) Confirmation response: the user's decisions are sent back to myIdP.
9) Attribute response: myIdP sends an attribute assertion back to the e-form provider.
10) Service: the e-form is displayed to the user and contains the selected and confirmed values from the SuisseID IdP/CAS and myIdP.

The user has now to complete the form and to enter the number of copies he wants to receive. In case, his email or home address has changed, he can also manually correct the data on the e-form (the data from the SuisseID IdP/CAS are read-only and can not be changed). When he saves the document the governmental process of providing the requested documents is started. But, the confirmed data from the e-form are also transferred – as new attribute assertions containing validated information – to myIdP (see Figure 5).

A crucial point to use myIdP in eGovernment applications and also in other domains is the selection and standardization of attributes. In our scenario, we could reuse attributes defined and published as Swiss standards, e.g., the eCH-0010 [20] for the address and eCH-0042 [21] for the email.

VI. CONCLUSION AND FUTURE WORK

myIdP is an extension to the SuisseID infrastructure. It proposes a Claim Assertion Service (SAML attribute authority), which handles personal data used and validated beforehand in other internet transactions. The concept is extensible to other eID solutions and can be also integrated in the STORK European eID Interoperability Platform. In a next step, the possibility to use myIdP as OpenID attribute provider will be investigated. Also the combination with a WebID seems feasible.

The myIdP concept was validated with a prototypical implementation following the proposed architecture. The implementation on the basis of the SuisseID SDK[22] showed quickly some limitations, especially related to a flexible attribute set and structured attributes, like address.

As proof-of-concept, the prototype was integrated in an eGovernment scenario of prefilling an e-form in order to obtain a proof of residence. The integration of more e-forms is planned. As precondition the set of myIdP attributes has to be extended to have a standardized basis for the information exchange.

The promoting of the myIdP service showed that many applications are willing to act as Attribute Requester and to use the personal attributes available in myIdP. The functionality to act as Claim Provider and to provide validated information to myIdP and to confirm the reuse is often seen as burden. But, both roles have to be equally provided to create a network of validated personal attributes.

Soon as more service providers will use myIdP and provide attribute claims, the model to calculate the assurance level can be validated on a real data basis and be further improved.

To strengthen even more the user-centric approach and to protect the private attribute, the central storage of claims in the myIdP database could be changed towards a pseudo-local approach that let the user choose where the store the data: on his own device or on a central place. The storage of SAML assertions on the user's device would also enable the usage of myIdP - in addition to the normal online scenario - in environments with limited or no connectivity.

ACKNOWLEDGMENT

Thanks to all students from the Bern University of Applied Sciences who helped with their bachelor theses [23][24] to realize this project.

REFERENCES

[1] Arbeitsgruppe Spezifikation des Trägerschaftsverein SuisseID, "eCH0113 SuisseID Specification, Version 1.5," November 30, 2011.

[2] Arbeitsgruppe SuisseID c/o Staatssekretariat für Wirtschaft SECO, "Claim Assertion Service (CAS), Technical Specification, Version 0.99.07," January 13, 2011.

[3] N. Ragouzis et al., "Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS Committee Draft," March 2008. [Online]. Available: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[4] J. Hodges, R. Philpott, and E. Maler, "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: https://www.oasis-open.org/committees/download.php/21111/saml-glossary-2.0-os.html

[5] M. Margraf, "The new German ID card," February 2011. [Online]. Available: http://www.personalausweisportal.de/SharedDocs/Downloads /EN/Paper_new_German_ID-card.pdf

[6] (2013, Mar) The official beID website. [Online]. Available: http://eid.belgium.be/en/

[7] (2013, Mar) Austrian Citiziens Card - Official Website. [Online]. Available: http://www.buergerkarte.at/index.en.php

[8] (2013, Mar) STORK - Project Website. [Online]. Available: www.stork-eid.eu

[9] (2013, Mar) STORK 2.0 - Project Website. [Online]. Available: www.eid-stork2.eu

[10] specs@openid.net, "OpenID Authentication 2.0 - Final," December 2007. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html

[11] (2013, Mar) Clavid - Official Website. [Online]. Available: clavid.ch

[12] (2013, Mar) Cloudid.de - OpenIDentity Provider - Website. [Online]. Available: cloudid.de

[13] Google, "Federated Login for Google Account Users," June 2012, accessed January 2013. [Online]. Available: https://developers.google.com/accounts/docs/OpenID

[14] D. Hardt, J. Bufu, and J. Hoyt, "OpenID Attribute Exchange 1.0 - Final," December 2007. [Online]. Available: http://openid.net/specs/openid-attribute-exchange-1_0.html

[15] Open AXN group. (2013, Mar) Street Identity - Website. [Online]. Available: https://sites.google.com/site/streetidentitylmnop/

[16] H. Story and S. Corlosquet (eds.), "WebID 1.0. Web Identification and Discovery. W3C Editor's Draft." January 2013. [Online]. Available: http://www.w3.org/2005/Incubator/webid/spec/

[17] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.98," August 2010. [Online]. Available: http://xmlns.com/foaf/spec/

[18] A. Keller, "Qualitätsmodell im Kontext von myIdP. CASE Arbeit." Master's thesis, BUAS - WGS, June 2012. [Online]. Available: http://www.myidp.ch/acms/fileadmin/documents/case_Qualitaetsmodell_v1.0.pdf

[19] S. Cantor, J. Moreh, R. Philpott, and E. Maler, "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[20] Verein eCH, "eCH-0010: Datenstandard Postadresse für natürliche Personen, Firmen, Organisationen und Behörden," October 2011. [Online]. Available: http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0010&documentVersion=5.00

[21] ——, "eCH-0042: Vorgehen zur Identifizierung von eGovernment-relevanten Geschäftsinhalten," June 2005. [Online]. Available: http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0042&documentVersion=1.00

[22] (2013, Mar) SuisseID SDK - website. [Online]. Available: https://www.e-service.admin.ch/wiki/display/suisseid/Home

[23] R. Imwinkelried and D. Ehrler, "Bachelorthesis: Specification myIdP," Master's thesis, BUAS - TI, January 2012.

[24] R. Bühlmann and M. Jeker, "Bachelorthesis: Specification myIdP Extensions," Master's thesis, BUAS - TI, June 2012.

A Service Localisation Platform

Luke Collins and Claus Pahl
School of Computing
Dublin City University
Dublin 9, Ireland
Email: luke.collins4@mail.dcu.ie, claus.pahl@dcu.ie

***Abstract*—The fundamental purpose of service-oriented computing is the ability to quickly provide software and hardware resources to global users. The main aim of service localisation is to provide a method for facilitating the internationalisation and localisation of software services by allowing them to be adapted to different locales. We address lingual localisation by providing a service translation using the latest web services technology to adapt services to different languages and currency conversion by using real-time data provided by the European Central Bank. Units and Regulatory Localisations are performed by a conversion mapping, which we have generated for a subset of locales. The aim is to investigate a standardised view on the localisation of services by using runtime and middleware services to deploy a localisation implementation. Our contribution is a localisation platform consisting of a conceptual model classifying localisation concerns and the definition of a number of specific platform services.**

Keywords - Service Localisation; Service-oriented Computing; Service-oriented Architecture.

I. INTRODUCTION

Distributed web services can provide business and private consumers with computing abilities which may not be feasible for them to develop in-house. These web services are currently in high demand in the context of cloud computing [3], [19]. However, the area of services computing introduces new issues, for example, in areas like Europe, where there is a wide range of languages spoken, services are very often only developed for single language and are only supported for that single language. Often it is the case that companies do not have the resources or capability to develop multilingual products. Localisation encapsulates a large number of issues which need to be addressed. These include, but are not limited to:

- Language Translation - conversion of services based on language. e.g., $English \rightarrow French$.
- Regulatory Compliance Constraints - conversion of services based on information such as taxation and other regulatory constraints.
- Currency Conversion - conversion of services based on currency, e.g., $Euro \rightarrow Dollar$.
- Units Conversion - based on standard units measurements, e.g., $Metric \rightarrow Imperial$.

Further concerns such as standardised vocabularies and conventions could be added.

Localisation is typically performed on textual content (i.e., strings) and refers to either languages only or physical location. However, the purpose of this work is to develop a method of localising services by implementing a 'mediator' type service which interacts between the Application Programming Interfaces (APIs) of the service provider and the requester. We are going to focus on a number of locale dimensions such as language, taxation, currency and units. An example of a request which requires localisation can be seen in Figure 1, which illustrates an example of a financial service provided to a range of locales (locations or regions requiring equal conversions).

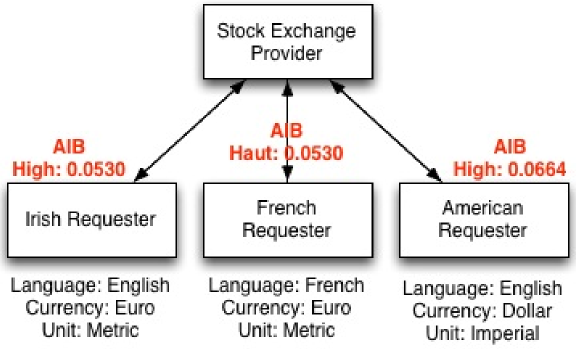


Fig. 1: Overview of Requests Requiring Localisation

We aim to provide service-level language translation techniques to localise services (including API interfaces) to different languages. Regulatory translation which includes currency, units and taxation among other legal governance and compliance rules will be provided by standards-based mappings. Regulatory translation is important for applications to comply with varying regional laws and regulations.

The objectives of service localisation include primarily the introduction of service-centric localisation techniques. A specific need is to make localisation techniques available at runtime for dynamic localisation, which is required for currencies and other variable aspects. Thus, Service Localisation (SL) provides a mechanism for converting and adapting various digital resources and services to the locale of the requester. A greater end-to-end personalisation of service offerings is an aim. A Localisation Provider act as an intermediary between the service provider and the requester. In our proposed platform, this is supported by a mediation service. By generating a common platform solution for these localisation issues, we allow the ability to dynamically localise Web services to be made with little effort. Our key contributions are:

- Software Localisation at Service Level - the main concern is a standardised mapping within a potentially heterogeneous environment.

- Adaptation and Integration - the main concern is the maintenance of service quality after it has been localised through adaptation.

The novelty of the proposed solution lies in filling a gap between service adaptation techniques (largely ignoring the regulatory and lingual aspects) and service internationalisation, which looks into localisation concerns, but only to a basic extent covering data formats and unit and currency conversions. We aim to show through a concrete example an appropriate use of Service Localisation. The example also attempts to illustrate various benefits and use cases. We also discuss motivating factors behind using a localisation technique.

In the next section, we discuss the motivation behind developing a Service Localisation implementation. Section 3 defines a platform architecture for Service Localisation. In Section 4, we introduce aspect-specific localisation techniques which we investigated and implemented. Section 5 introduces the implementation and evaluates our solution to the Service Localisation problem. Section 6 contains the related work discussion. In Section 7, future directions and possible extended infrastructures are explored. We end with related work and concluding comments.

II. Motivation

Our main focus is a platform for service localisation, which makes a shift from the typical "one size fits all" scenario towards a more end-to-end personalised service scenario. Currently, services computing suffers from localisation and adaptability issues for multiple users in different regions. These issues could be overcome if a multi-lingual and multi-regional solution was developed [15], [22]. The different localisation issues of a service can be illustrated. The scenarios described below are used to illustrate benefits to service localisation:

- End-User Services: Some software-as-a-Service providers only support one region with one specific language. There is a possibility to perform localisation both statically (compile-time) and dynamically (runtime), which typically involves localising service values and interacting messages.
- Business Services: Various business centric applications including applications for documentation and analysis could be localised to support various legal and regional locales. Business services typically require more customisation than end-user consumers.
- Public Sector Services: As governments outsource their computing infrastructure to external providers, it is becoming more important for the providers to supply solutions which take into account various regulatory governance aspects such as currency and taxation and also lingual localisation.

Another scenario which provides a detailed view of the benefits of service localisation could be a service provider, used to manage company accounts for its customers. This could be a company which has offices in different global locations and would like to provide localisation based on customer region and localisation for its individual offices.

- Regulatory: Conversion of data between standards and their variants, e.g., based on different units of measurement $Metric \rightarrow Imperial$.
- Currency: Conversion of between currencies, e.g., $Euro \rightarrow Dollar$.
- Lingual: Translation of service related data between languages. This could include free text, but also specific vocabularies based on business product and process standards such as GS1 or EANCOM.
- Taxation: Different customers have different taxation requirements, e.g., VAT rates. Localisation of accounts software can take this into account for each locale.

A. Use Cases and Requirements

In order to demonstrate the need for localisation of Web services, we chose to demonstrate the issue using a concrete case of an environment which utilises service-level access to a stock exchange interface. Imagine an Irish user who wishes to access data from the New York Stock Exchange, which is provided in an English format with the currency in dollars. A user in France may also wish to access data from the New York Stock Exchange using a French interface where local regulations require French to be used for data and/or service operations. Therefore, there must be a mechanism to convert the currency to Euro or to another currency which the requester specifies. There must also be a mechanism to convert the language to that of the requester.

At application level, two sample calls of a stock exchange data retrieval service for the two different locales (IE-locale with English as the language and EUR as the currency and FR-locale with French as the language and EUR as the currency) retrieve the average stock price for a particular sector - in this case the financial sector as follows:

- $Retrieve(20/08/2012, Financial) \rightarrow 30.50\, EUR$
- $Récupérer(20.08.2012, Financier) \rightarrow 30,50\, EUR$

In the US-locale with English as the language and USD as the currency, the same API call could be the following:

- $Retrieve(08/20/2012, Financial) \rightarrow 38.20\, USD$

The following elements in this case are localisable:

- Date: in order to preserve regulatory governance, the date format requires to be changed depending on the requester locale.
- Language: names of functions from the API are translated between languages.
- Currency: values are converted as normal and this would apply to any other units.

This list can vary depending on the environment where different regulatory constraints might apply. In general, it can be expected that there is always a linguistic element to the localisation of any product, but elements may also include taxation and units of measurement. If it was the case that the requesters were trying to access weather forecasts for their

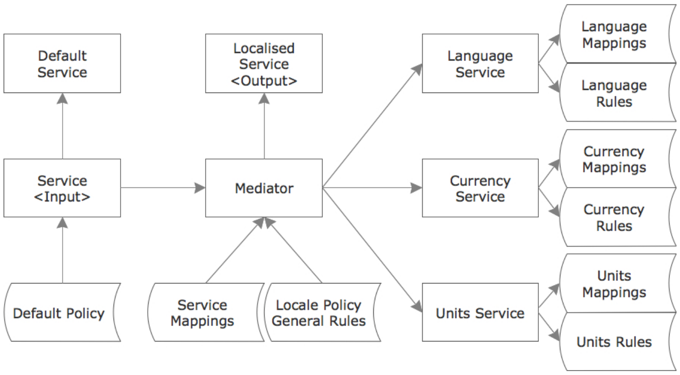


Fig. 2: Architecture of the Localisation Platform.

own region and formatted in their own locale, then it would be necessary to utilise a conversion for units of measurement:

- $Prévision(20.08.2012) \rightarrow 30° Celsius$
- $Forecast(20/08/2012) \rightarrow 15° Celsius$

In the US-locale with English and imperial units, the same API call could be $Forecast(08/20/2012) \rightarrow 87° Fahrenheit$.

III. Platform Architecture

Localisation of services requires a framework to be implemented to facilitate various localisation methods. These various methods, implemented as services in our proposed localisation platform, are used to facilitate the localisation of localisable elements or artefacts. This paper focuses on the dynamic localisation of service level descriptions.

With every service there are various elements which may be localised. These elements include:

- Service specifications/descriptions (APIs)
- Models (structural/behavioural)
- Documentation (for human consumption)
- Messages exchanged between services

Services are normally written to be independent of locales, however localisation is often needed. A localisation platform should be based on attributes which vary from locale to locale, like time or date format.

A service localisation platform requires a number of elements. These elements can be pre-translated fragments in static form or can be dynamic translation systems. Figure 2 aims to demonstrate the concept of a policy and mappings based system, which can be scaled when additional processes are attached to the mediation process. In the platform architecture, user-specific locale policies are applied to service endpoints. For example, in a WSDL file we may localise messages and operation names. Rules for each type of translation would be stored in a rules database (General Rules Repository). Similarly, mappings between common translations would be stored in a mappings database (Translation Memory).

A mediator operates between users (with different locales) and several service providers (with different locales) by providing core localisation services, such as currency conversion and language translation. The architecture supports the following:

- Static Mappings: these could be the mapping of one language to another or one unit to another, pre-translated in translation memories.
- Dynamic Localisation: when translation mappings are not stored, dynamic localisation is required in order to obtain a correct translation and store the mapping.
- Policy Configuration: in order to configure the various locale policies, we must generate particular translation rules, supported by a logical reasoning component.
- Negotiation: this is the exchange of locale policies through the form of XML and SOAP from a web services point of view.
- Localisation of Services: the mappings between the remote service and the localised service description must be stored in a mappings database (Translation Memory) so the localised service has a direct relationship with the remote service.

The workflow is concequently $Negotiation \rightarrow PolicyConfiguration \rightarrow Localisation \rightarrow Execution$.

Some examples shall illustrate the functionality of the platform. Table I defines two different locales in XML profiles. A mismatch between the requester locale and the provider locale needs to be bridged by the mediator localisation service. The language as a lingual aspect and country, currency and unit codes are regulatory concerns.

TABLE I: Sample Environment Setup

```
<SLContext>
   <Locales>

      <RequesterLocale>
        <LanguageCode>efr </LanguageCode>
        <CountryCode>FR</CountryCode>
        <CurrencyCode>EUR</CurrencyCode>
        <UnitCode>M</UnitCode>
      </RequesterLocale>

      <ProviderLocale>
         <LanguageCode>en </LanguageCode>
         <CountryCode>IE </CountryCode>
         <CurrencyCode>EUR</CurrencyCode>
         <UnitCode>M</UnitCode>
      </ProviderLocale>

    </Locales>
  </SLContext>
```

The locale definitions decide how a given service API (in WSDL) is localised. Results from a sample execution of the localisation service (the mediator) is displayed in Tables II and III based on the XML locale definitions of the environment in Table I. Table II shows excerpts from an original WSDL file. Table III shows the localised WSDL after the application of lingual localisation in this case (translation from English (IE locale) into French (FR locale) – for simplicity of the example, we have focused on this single aspect only), compliant with the two locale definitions from the first listing.

TABLE II: Sample Input - Provider Locale

```
<wsdl:message name=”quoteResponse”>
  <wsdl:part name=”parameters”
             element=”quoteResponse”/>
</wsdl:message>
<wsdl:message name=”quoteRequest”>
  <wsdl:part name=”parameters”
             element=”quote”/>
</wsdl:message>
<wsdl:portType name=”Quote”>
  <wsdl:operation name=”getQuote”>
    <wsdl:input name=”quoteRequest”
                message=”quoteRequest”/>
    <wsdl:output name=”quoteResponse”
                message=”quoteResponse”/>
  </wsdl:operation>
</wsdl:portType>
```

TABLE III: Sample Output - Localised WSDL

```
<wsdl:message name=”quoteReponse”>
  <wsdl:part name=”parameters”
             element=”quoteReponse”/>
</wsdl:message>
<wsdl:message name=”citerDemande”>
  <wsdl:part name=”parameters”
             element=”citer”/>
</wsdl:message>
<wsdl:portType name=”Citer”>
  <wsdl:operation name=”getCiter”>
    <wsdl:input name=”citerDemande”
                message=”citerRequest”/>
    <wsdl:output name=”citerDemande”
                message=”citerDemande”/>
  </wsdl:operation>
</wsdl:portType>
```

IV. Localisation Platform – Rules and Services

We have outlined the core platform architecture in the previous section with the central services. In order to provide the localisation platform services, we need to implement a number of classes to enable a modular service localisation platform. Their interaction is summarised in Figure 3. Details of underlying concepts of their operation are explained now.

A. Rule-based Locale Definition and Conversion

At the core of our service localisation platform is a language to specify the rules in relation to localisations. In most cases, languages like WSDL and other XML languages provide information regarding the services that are provided via an API. However, in order to encapsulate localisation information, there is a necessity to provide a language which will contain details in relation to the locales of the requester and the provider. For the purpose of our localisation platform, we use a policy language based on the Semantic Web Rule Language SWRL, which is based on the propositional calculus.

A localisation layer encapsulates the various forms of translations. It describes the relationships between localisable elements. For example, it contains the details of items which can be translated. For our localisation model these are documentation and descriptions, but also API messages and operations. The rule language is used to define policies of two types: firstly, locale definitions and, secondly, conversion (translation) rules. We motivate the rule set through examples.

Firstly, there are a number of locale definition rules provided, like *Loc* or *hasCur*, by which locales for specific regions are described. A locale can also be described by other rules such as *hasTax*, *hasLang* and *hasUnit*. Examples of three region's locales - IE, US, and FR - are:

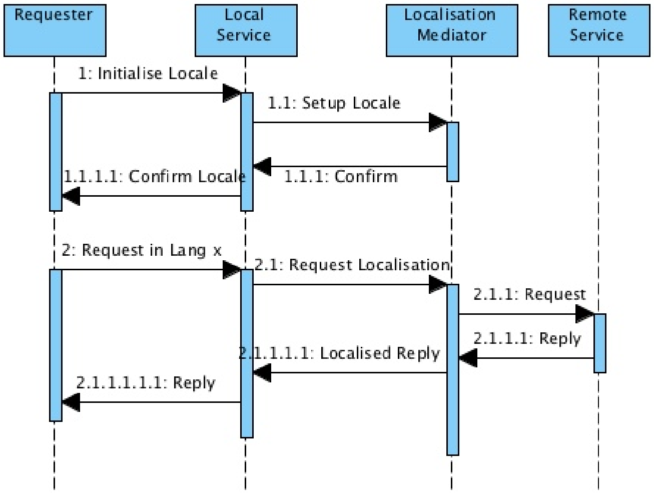


Fig. 3: A UML Sequence Diagram of the Platform.

$$\begin{aligned}&IELoc(?l) \leftarrow Loc(?l) \wedge\\&\quad hasLang(?l,?z) \wedge hasCur(?l,?c) \wedge hasUnit(?l,?u) \wedge\\&\quad ?z = en \wedge ?c = EUR \wedge ?u = metric\end{aligned}$$

$$\begin{aligned}&USLoc(?l) \leftarrow Loc(?l) \wedge\\&\quad hasLang(?l,?z) \wedge hasCur(?l,?c) \wedge hasUnit(?l,?u) \wedge\\&\quad ?z = en \wedge ?c = USD \wedge ?u = imperial\end{aligned}$$

$$\begin{aligned}&FRLoc(?l) \leftarrow Loc(?l) \wedge\\&\quad hasLang(?l,?z) \wedge hasCur(?l,?c) \wedge hasUnit(?l,?u) \wedge\\&\quad :?z = fr \wedge ?c = EUR \wedge ?u = metric\end{aligned}$$

The benefit of a formal framework for the rules is that other rules can be inferred by from partial information. For example, if we knew that a locale had USD as its currency we may be able to infer its country from it:

$$?c = USD \rightarrow ?l = USLocale.$$

These inferred rules do not apply in general - this may not work if we know a currency is Euro in which case it could be one of many locales in Europe. The purpose of these rules could be to determine inconsistencies, however. Preconditions can clarify the remit of these rules.

Secondly, a generalised conversion between locales, e.g., $Locale\ A \rightarrow Locale\ B$, is given by the following general conversion rule:

$$\begin{aligned}&IELoc2USLoc(?l1,?l2) \leftarrow\\&\quad hasLang(?l1,?z1) \wedge hasLang(?l2,?z2) \wedge\\&\quad hasCur(?l1,?c1) \wedge hasCur(?l2,?c2) \wedge\\&\quad hasUnit(?l1,?u1) \wedge hasUnit(?l2,?u2) \wedge\\&\quad ?z2 = convertLang(en,en,?z1) \wedge\end{aligned}$$

$$?c2 = convertCur(EUR, USD, ?c1) \land$$
$$?u2 = convertCur(metric, imperial, ?u1)$$

$$IELoc2FRLoc(?l1, ?l2) \leftarrow$$
$$hasLang(?l1, ?z1) \land \ hasLang(?l2, ?z2) \land$$
$$hasCur(?l1, ?c1) \land \ hasCur(?l2, ?c2) \land$$
$$hasUnit(?l1, ?u1) \land \ hasUnit(?l2, ?u2) \land$$
$$?z2 = convertLang(en, fr, ?z1) \land$$
$$?c2 = convertCur(EUR, USD, ?c1) \land$$
$$?u2 = convertCur(metric, metric, ?u1)$$

Depending on requester and provider locale any combination of mappings/translations can be generated by the core rules.

B. Localisation Mediator

Based on these local definition and conversion rules, a number of services operate. In order to provide a transparent localisation system, one class acts as a mediator, as visualised in Figure 4, which could use individual services for: Lingual Conversion, Currency Conversion, Regulatory Governance, Units Conversion, and WSDL Parsing & Generation. Within this mediator, implemented as a Java class, we have various methods which call the other localisation services of the platform.

During execution of the localisation platform, an XML file is first passed to the mediator. The Mediator Service then sets up a localisation environment using the locale details provided in *LocaleConfig.xml*, the class performs this via the use of Java Interfaces. Once the locale is set up, the service Web Service Description Language (WSDL) file is parsed and various elements are localised resulting in a localised WSDL file which can be used to access localised operation mappings. This class is the work horse of the platform and can be extended with the introduction of other localisation classes.

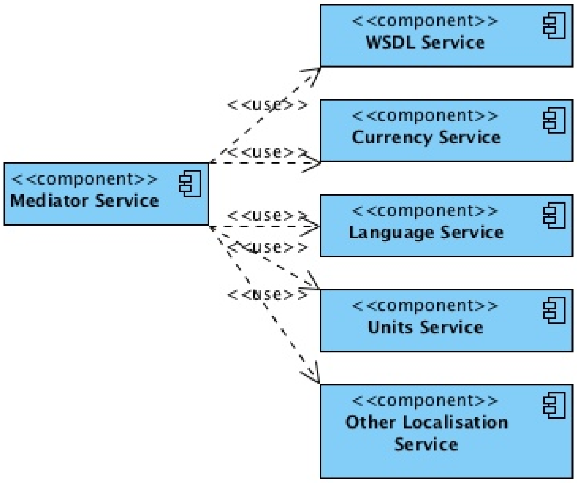


Fig. 4: A Component Diagram Displaying Extensibility.

Linguistic artefacts are one of the most broadly localised elements of software today. We propose machine translation (MT) to achieve automation. While further research into a tailored MT solution is required to specifically address limited textual context and controlled vocabularies for APIs, language translation within the proposed platform is provided by the Google Translate API. In the interest of performance, our platform tries to make as few API calls to Google as possible. Instead it stores translations of popular words and glossaries within a local language mapping database (Translation Memory) for later retrieval. A local machine translation system may also reduce this latency, as it would no longer have to depend on TCP/IP performance. The conversion rule for language translation is given by:

$$IELoc2FRLoc(?l1, ?l2) \leftarrow$$
$$hasLang(?l1, ?z1) \land \quad hasLang(?l2, ?z2) \land$$
$$?z2 = convertLang(en, fr, ?z1)$$

Regulatory localisation through adaptation to other regulatory standards is based on localising regulatory concerns. These concerns include, but are not limited to the following: Taxation, Currency, and Units of Measurement. We have chosen to localise a subset of these concerns. For the purpose of units localisation, we developed an interface to a repository of unit conversion formulae. These formulae provided conversions between the metric and imperial units of measure. The conversion rule for units is given by:

$$IELoc2USLoc(?l1, ?l2) \leftarrow$$
$$hasUnit(?l1, ?u1) \land \ hasUnit(?l2, ?u2)$$
$$\land\ ?c2 = convertUnits(metric, imperial, ?u1)$$

Due to a large number of currencies used globally, it was necessary to develop a class which dealt with currency conversion. For the purpose of currency localisation, we use exchange rates from the European Central Bank. This is in our case supported by a MySQL database. Currencies are manipulated based on their rate compared to Euro as the base currency. The conversion rule for currency is given by:

$$IELoc2USLoc(?l1, ?l2) \leftarrow$$
$$hasCur(?l1, ?c1) \land \ hasCur(?l2, ?c2)$$
$$\land\ ?c2 = convertCur(EUR, USD, ?c1)$$

In order to parse the input in the form of WSDL files, a WDSL service class is used. This class contains the methods required to manipulate both incoming WSDL files of the service provider and has the ability to generate a localised WSDL file. The class can be considered as an I/O Manager. XLIFF is an XML standard for translation that proved useful when it comes to the localisation of WSDL file.

V. Implementation and Evaluation

The localisation platform presented here was fully implemented in a Java prototype that aims at studying the feasibility of the conceptual solution. It shall be assessed on the following criteria here: Performance and Extensibility. These criteria have different effects on the end-user experience of the product. These criteria are key performance indicators (KPI) and critical success factors (CSF) of the localisation platform described.

Poor performance often tends to affect software exponentially as multiples of users consume a service at the same time. The core question here is the overhead created by adding localisation dynamically to service provisioning. Our results show an acceptable overhead of 10-15 % additional execution time for fully localised services (i.e., localisation involving different localisation aspects). The overhead is still low compared to network latency and the average service execution time [22]. As the application deals with multiple users, the latency would increase due to extra loads placed on the platforms services. This makes latency one of the KPIs of the project. Latency, is also an area to be assessed as adding the localisation platform to the workflow of an existing process has the potential to add lag-time. This lag-time exists due to time required to compute and also the time to initialise the various variables. The propagation latency is displayed in Table IV below. It should be noted that figures can be affected by environmental changes or the locale we are transforming from and the locale we are transforming to.

TABLE IV: Latency Table - Localisation of Service

Service	Prior (μs)	Post (μs)	Δt (μs)
NASDAQ	132	182	50
FTSE	110	152	42

As a general strategy, we have aimed to improve performance by using pre-translated aspects (through stored mappings). A related concern is scalability of software becomes more important when a service may have large multiples of users. Scalability has not been empirically addressed for this phase of research and will be evaluated in later prototypes.

- Some components of the platform would require modification to effectively allow the infrastructure to vertically scale-up or scale-out efficiently. Solutions here are stateless programming and data externalisation. Through our rule base, and the suggested pre-translation repositories some suitable architectural decision in this direction have already been made.
- Horizontal scalability - i.e., the addition of more localisation concerns - is conceptually easily supported by the modular mediator architecture, which we will address further below in the extensibility context from an implementation view.

An interesting model to investigate the scalability is a tuple space-based coordination approach [6], [7], [11], which would allow a flexible and elastic assignment of localisation services to multiple requests.

Extensibility becomes important when dealing with complete platforms like a localisation platform. During an initial development, it is often the case that features need to be included due to various constraints. In the case of the localisation platform described here, some localisation services where not developed, some of which include a service to handle taxation. However, the platform was designed to be extendable. At a platform level, this allows for the addition of further services and the support for more locales.

VI. Related Work

We provide a different view and perspective on the subject compared to other publications [9], [15], [18]. The area of localisation in its wider adaptivity and customisation sense has been worked on in various EU-supported research projects, such as SOA4ALL [17] and 4Caast [**?**]. These projects address end-user adaptation through the use of generic semantic models. Areas such as software coordination are also covered. The mOSAIC project adds multi-cloud provision to the discussion. Our framework however is modular and extensible and aims to provide a one-stop shop for all localisation methods.

The platform which is described here addresses the need for dynamic localisation of various artefacts by use of a translation memory and a set of logical rules. Software Localisation refers to human consumption of data which are produced by the software - namely messages and dialogues. Our focus is on the localisation of the service level. Service internationalisation is supported by the W3C Service Internationalisation activity [16], [20]. Adaptation and Integration of services based on locales and using a translation memory with rules and mappings is new [18]. The problem of multi-tenancy is a widespread issue in the area of cloud computing [22]. This is an area where a lot of research is being invested in order to provide a platform for different users with different business needs to be kept separate and their data to be kept private. Semantics involves the matching of services with various locales using mappings and rule-based system [2], [4], [9].

There are implementations which can perform localisation operations on web services [10]. The use of some of these however is restricted due to the nature of them. Some of the other implementations require a specific Integrated Development Environment or specific proprietary libraries. They also typically enable localisation at compile time - the proposed implementation in this paper is to enable service localisation at run time. IBM has presented a static localisation solution suitable for web services using its WebSphere platform [10], which requires the WSDL files to be generated within the Integrated Development Environment prior to deployment. This differs from our proposed localisation platform as our solution aims to perform transformations between locales dynamically.

VII. Conclusion and Future Work

Service localisation falls into the service personalisation and adaptation context. There are particular engineering methods and tools which can be employed to allow services to be adapted to different locales. A Service Localisation implementation should allow for automatically adjusting and adapting services to the requesters' own locales. We have presented a modular implementation which can enable services to be introduced into emerging markets which have localisation issues. Localisation hence provides a mechanism to widen a service provider's target market by enabling multi-locale solutions. The easiest solution is for a service provider to provide a 'mediator' service which could act as middleware between a requester and the service provider.

By allowing services to be localised, we are enabling the provision of multi-locale services to create interoperable service ecosystems (such as clouds). Due to the nature of third-party services, it is more intuitive for service localisation to be performed dynamically through the use of a mediator service. Service localisation thus enables higher availability of services through its use of innovative interfacing. This type of localisation would be value-add for a company which may not have the resources to perform localisation in-house.

The objectives of Service Localisation have been presented in two forms. Firstly, presented was a conceptual framework which demonstrated key motivational reasons for developing a multi-locale support framework. The second part presented a modular platform, which is extensible to allow the support of further localisable artefacts. The platform which was implemented was using Java libraries was discussed as this programming solution copes well with the problem of extensibility.

The proposed service localisation fills a gap. Software adaptation has looked into adapting for instances services in terms of their user's interface needs such as data types and formats. The two focal localisation concerns lingual and regulatory add new perspectives to this area of research. A different activity is the Web services internationalisation effort, which looks into basic localisation concerns such as units, currency or the format of dates. Our localisation solution includes these (as we have demonstrated with the currency aspect), but expands these into a comprehensive framework.

The context of adaptation and translations/mappings used to facilitate this is a broad field. Our aim here was to integrate difference concerns into a coherent localisation framework. This relies on individual mappings. As part of our future work, we aim to add a semantic layer, which would support to concerns. Firstly, it would allow more reliable translations for non-trivial concerns if overarching ontologies were present. Secondly, the different concerns themselves could be integrated by determining interdependencies. Another direction of future research is to look into composition and, specifically, the behaviour of individual service localisation in for instance service orchestrations or other coordination models (e.g., tuple spaces as suggested above).

REFERENCES

[1] 4CaaSt. ”Building the PaaS Cloud of the Future”. *EU FP7 Project*. http://4caast.morfeo-project.org/. 2013.

[2] D. Anastasiou. ”The impact of localisation on semantic web standards.” *European Journal of ePractice*, 12:42–52. 2011.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin and I. Stoica. ”A view of cloud computing.” *Communications of the ACM*, 53(4):50–58. 2010.

[4] K.Y. Bandara, M.X. Wang and C. Pahl. ”Dynamic integration of context model constraints in web service processes.” *International Software Engineering Conference SE'2009*. IASTED. 2009.

[5] K. Chen and W. Zheng. ”Cloud computing: System instances and current research.” Second International Conference on Future Networks, 2010. ICFN '10, pp. 88–92. 2010.

[6] G. Creaner and C. Pahl. ”Flexible Coordination Techniques for Dynamic Cloud Service Collaboration.” In Proceedings Workshop on Adaptive Services for the Future Internet WAS4FI. ServiceWave 2011. 2011.

[7] E.-E. Doberkat,W. Hasselbring, W. Franke, U. Lammers, U. Gutenbeil, and C. Pahl. ”ProSet - a language for prototyping with sets.” In International Workshop on Rapid System Prototyping 1992. pp. 235-248. IEEE, 1992.

[8] P. Fingar. ”Cloud computing and the promise of on-demand business inovation.” *InformationWeek*, July 13, 2009.

[9] K. Fujii and T. Suda. ”Semantics-based context-aware dynamic service composition.” ACM Transactions on Autonomous and Adaptive Systems (TAAS), 4(2):12. 2009.

[10] IBM. ”IBM Developer Technical Journal: Developing internationalized Web services with WebSphere Business Integration Server Foundation V5.1.” 2010.

[11] C. Pahl. ”Dynamic adaptive service architecturetowards coordinated service composition.” In European Conference on Software Architecture ECSA'2010. pp. 472-475. Springer LNCS. 2010.

[12] C. Pahl. ”Layered Ontological Modelling for Web Service-oriented Model-Driven Architecture.” *European Conference on Model-Driven Architecture - Foundations and Applications ECMDA'05*. Springer. 2005.

[13] C. Pahl, S. Giesecke and W. Hasselbring. ”An Ontology-based Approach for Modelling Architectural Styles.” European Conference on Software Architecture ECSA'2007. Springer. 2007.

[14] C. Pahl, S. Giesecke and W. Hasselbring. ”Ontology-based Modelling of Architectural Styles.” Information and Software Technology. 1(12): 1739-1749. 2009.

[15] C. Pahl. ”Cloud Service Localisation.” European Conference on Service-Oriented and Cloud Computing ESOCC 2012. Springer. 2012.

[16] A. Phillips. ”Web Services and Internationalization.” Whitepaper. 2005.

[17] SOA4All. ”Service Oriented Architectures for All”. *EU FP7 Project*. http://www.soa4all.eu/. 2012.

[18] H. Truong and S. Dustdar. ”A survey on context-aware web service systems.” Intl Journal of Web Information Systems, 5(1):5–31. 2009.

[19] W. Voorsluys, J. Broberg and R. Buyya. ”Cloud Computing: Principles and Paradigms.” John Wiley and Sons. 2011.

[20] W3C. ”Web Services Internationalization Usage Scenarios.” W3C. 2005.

[21] M.X. Wang, K.Y. Bandara and C. Pahl. ”Integrated constraint violation handling for dynamic service composition.” IEEE Intl Conf on Services Computing. 2009. pp. 168-175. 2009.

[22] M.X. Wang, K.Y. Bandara and C. Pahl. ”Process as a service distributed multi-tenant policy-based process runtime governance.” International Conference on Services Computing (SCC), pp. 578–585. IEEE. 2010.

[23] H. Weigand, W. van den Heuvel and M. Hiel. ”Rule-based service composition and service-oriented business rule management.” Proceedings of the International Workshop on Regulations Modelling and Deployment (ReMoD'08), pp. 1–12. 2008.

A Service Component-Oriented Design and Development Methodology for Developing SOA-based Applications

Soumia Bendekkoum, Mahmoud Boufaida
LIRE Laboratory
Mentouri University of Constantine
Constantine, Algeria
{soumia_bendekkoum,mboufaida}@umc.edu.dz

Lionel Seinturier
LIFL Laboratory & INRIA Lille
University Lille 1, Villeneuve d'Ascq
Lille, France
Lionel.Seinturier@univ-lille1.fr

***Abstract*—The Service-Oriented Architecture (SOA) is a promising technology based standard for easily developing distributed, interoperable and loosely coupled applications. The emergence of the Service Component Architecture (SCA) standard, which uses service components, has more and more facilitated the development and deployment of SOA independently from technologies and standards. However, SOA does not define a complete and clear methodology for developing service based systems. It does not address the issue of the way these services could be defined despite the fact that there exist successful specification tools (UML profiles, Service Component) for modeling service-based applications. This paper presents the Service Component-oriented Design and Development Methodology (SCDD-Methodology), which combines software engineering approach and service component models to specify and indentify adequate services. It discusses the key principles in its design: the adoption of service component model for the development of SOA-based applications for well defining the structure of the application behind the service's layer. The paper presents a case study of a commercial company producing machine tools (MTP).**

Keywords-SOA; SCA; UML Profiles Modeling approaches; Service Component.

I. INTRODUCTION

SOA is an architectural style for the reorganization and redeployment of the information system [1]. It encapsulates the functions of an application into a set of loosely coupled services. These services are defined with a contract, and published using an interface description, so that they can be invoked by remote clients.

SOA is not only a technology or a recipe. It is a way of thinking and structuring distributed information systems. It requires appropriate modeling tools and good methodologies for the design, development and management of distributed applications conforming to its principals of autonomy, reuse, interoperability and loose coupling of its different elements. Despite the success of SOA, it remains a partial solution [2], [3], because it describes how application's functionalities are structured into autonomic and distributed entities (services) as well as how they are published and used on the web. But it does not define what is behind the scene and how these services are structured [4]. The recently developed Service Component Architecture (SCA) is proposed to fulfill that shortcoming with a set of specifications, which supports a view of service as software component. This means that behind the service layer there exists a set of components named "Service components" implementing service behavior. Several platforms have already been developed that implement SCA specification, such as Tuscany [5], Newton [6] and FraSCAti [7] for java-based SCA applications.

SOA development and implementation methods are primarily different in their details, but they all have the same principal, which is according to Heubès [8] in the same spirit as the engineering business processes or information systems. Zimmermann et al. [9] and Papazoglou and Heuvel [10] reveal that there exist three methods for developing an SOA. The first one is the *Top-down* in which the business logic of existing processes is used to identify services. In contrast, *Bottom*-up approach starts with the analysis of applications to determine the existing functions of the information system, and from these artifacts, it is possible to identify the functions that are eligible to the level of service. Finally, the hybrid approach called also *Meet in the Middle* approach advocates to conduct in the same time a top-down and bottom up methods. Therefore, the main steps of an SOA development approach can be summarized as follows [11]: (1) business processes are represented, (2) analyzed, (3) improved and (4) built on the top of the existing legacy applications.

Unfortunately, although that there exist successful specification tools for realizing a service-oriented development project [12], these specification approaches do not rely on an effective development methodology [3], especially for complex service-oriented architecture projects, where we need enormously. Our studies on existing SOA development projects reveal that these approaches do not provide a comprehensive, clear and precise strategy to achieve a successful SOA development project. In consequence, the developers are usually facing a major difficulty, which is the definition of the concept of service. This problem causes to the companies an important question which is what a service is, and what level of granularity can it takes to properly define the relevant services to a business based integration project.

In this paper, we present the Service Component-oriented Design and Development Methodology (SCDD-Methodology), which combines software engineering approach and service component models to specify and construct as well as structured SOA-based applications.

This paper is organized as follows. Section 2 presents the SCA specification standard used in the SCDD methodology, and discusses its key principals for achieving a service oriented development project. Section 3 describes the service component design and development methodology. Section 4 presents a case study of a commercial company producing machine tools and describes some implementation aspects. Section 5 discusses related work. Finally, Section 6 concludes the paper and outlines some future works.

II. SCA MODELING APPROACH AND SOA CHALLENGES

The SCA is a set of specifications defining component model for building Service-based applications using SOA and component-based software engineering (CBSE) principals.

SCA entities are software components, which may provide interfaces (called services), require interfaces (called references) and expose properties. References and services are connected through wires (Figure 1.).

These models allow us to offer specification tools, which facilitate the modeling and implementation of architectures using services as well as a flexible layer for application interobperability. So that, these models have self-configuration capacities that permit to cope with the continuous changes of both the environment and client needs.

Service component models have important properties that motivate our choice of research. We summarize those which are related to our work, as follows [14]:

- Hierarchy: the SCA model is hierachical with components being implemented by primitive language entities or by subcomponent
- Autonomy: The service component architecture model is equipped with the autonomy feature, which is useful to define the functions of components that implement business services inside or outside of the grid services of the model architecture.
- Reconfiguration: service component models are equipped with an important property called self-reconfiguration or dynamic reconfiguration [15], which can be useful to ensure effortlessly the adaptability of SOA business services. For example, it is conceivable that any change in the definition of a requested service can be easily mapped in a reconfiguration of Fractal component model [16]. This is due to the interfaces (internal and external) and also the property of the plug and play between components of the hierarchical model.

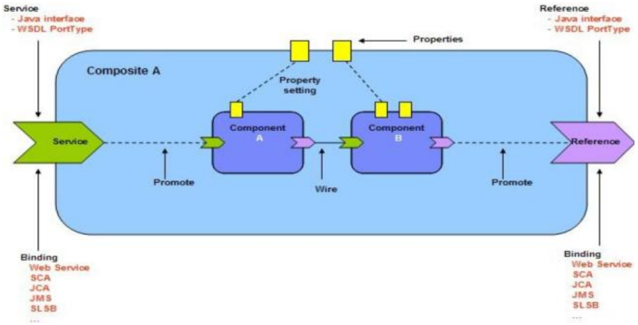


Figure 1. Exemple of SCA Component Architecutre [13].

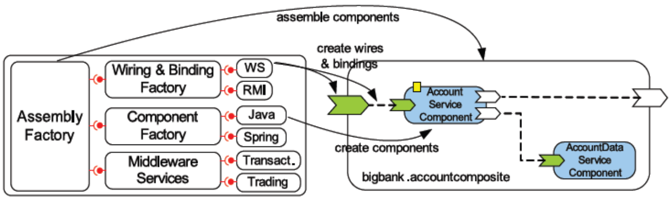


Figure 2. FraSCAti Platform Architecture [13].

SCA is based on the idea that business function is provided as a series of services, which are assembled together to create solutions that serve a particular business need. These composite applications can contain both new services created specifically for the application and also business function from existing systems and applications, reused as part of the composition. SCA provides a model both for the composition of services and for the creation of service components, including the reuse of existing application function within SCA compositions [17], and this motivates more our choice of using component model.

SCA aims to encompass a wide range of technologies for service components and for the access methods, which are used to connect them. For components, this includes not only different programming languages, but also frameworks and environments commonly used with those languages. For access methods, SCA compositions allow for the use of various communication and service access technologies that are in common use, including, for example [13], Web services, messaging systems and Remote Procedure Call (RPC).

We present, as an example, the SCA FraSCAti platform, which we are chosing for the deployment of the service component-based applications. The platform has four main components, as shown in Figure 2: *Component Factory*, *Wiring and binding factory*, *Middleware services,and Assembly factory* [13].

In the next section, we detail the various phases of the proposed approach, each of which is illustrated with an exemple.

III. PHASES OF THE SCDD METHODOLOGY

The objective of the service component-oriented design and development methodology is to achieve service integration and to facilitate service interoperability as well as service composition. In our approach we focus on the SCA concepts (autonomy and composition) for achieving our objective to create an SOA-based system, which responds more to the client requirements and to well structure the business logic of its functionalities. The SCDD methodology provides appropriate principals and guidelines to specify, construct and customize well defined services (fine grained functions) and business processes (coarse grained functions) orchestrated from fine grained services (service composition).

As shown in Figure 3, the SCDD-Methodology constitutes four main phases: (i) Modeling phase: Modeling Business Needs and Modeling Existing Applications, (ii) Service Components Identification, (iii) Service Components deployment, and finally, (iv) Service Identification and Publication phase. In the following section, we detail separately each phase:

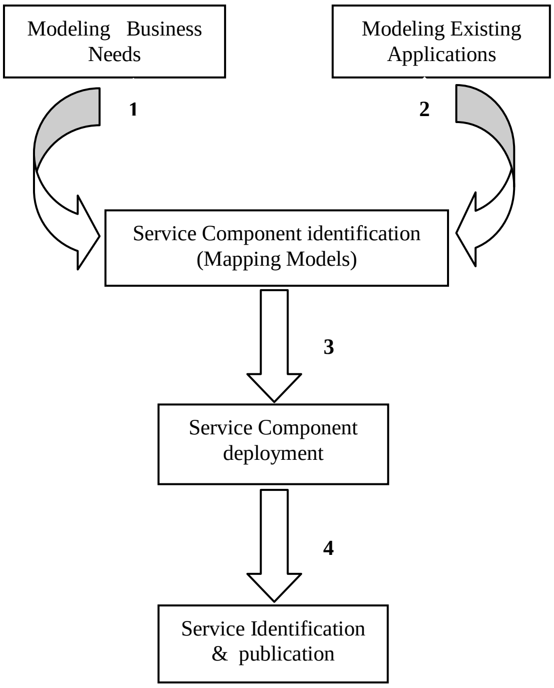

Modeling Business Needs
Modeling Existing Applications
1
2
Service Component identification (Mapping Models)
3
Service Component deployment
4
Service Identification & publication

Figure 3. Principal Phases of SCDD Methodology.

A. Modeling Phase:

The first phase of SCDD methodology consists of analyzing and modeling the requirements of the new system. This includes studying and modeling existing applications. This phase contains two steps, the first one is Business Needs Modeling for analyzing and modeling clients' requirements, and the second is Existing Applications Modeling for analyzing and modeling applications of the enterprise legacy system:

1) *Business Needs Modeling:* objectives are business goals which custumors want to achieve, or target behaviors which customers want to see in the system. It is important when setting objectives to make sure they are in line with overall business needs of the target information system, this is the principal objective of our approach: the allignement of the business services of the futur system to the client business needs. To do so, we propose, in this first step, to analyze the business goals of the future information system, and model these goals in an Objective Tree Model.

Figure 4 represents an UML metamodel defines the structure of an Objective Tree Model. This model represents a hierarchical structure specification of all needs of the future SOA-based system. Each node of the tree represents an objective and its sub-nodes represent sub-objectives realizing the overall objective. Constraints are functional conditions, which must be supported by an objective to achieve an other objective, these constraints are very important in the objective tree model, to identify to the designer and, in the future, to the developer any conditions might make echeiving the objective more diffecult or even in some cases impossible.

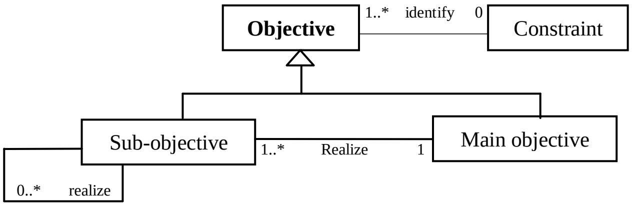


Figure 4. UML Meta-model of Objective Tree Model

The tree model obtained, in this step, is crucial to the alignment of the business process of the target SOA-based system to the client and environment business needs. This model will be used in the follwing phase for analyzing and defining the components that permit to implement the adequate functions that best respond to the company's needs.

2) *Existing Applications Modeling:* In SOA-based systems business processes are modelled in different granularities, these processes incorporate functions (services) supported by the existing enterprise legacy system, as well as by new functions not supported by the existing system, which must be developed. In this step, we identify adequat services in the right granularity. To do so, our approach proposes the use of Service Components Models, to specify existing legacy systems. These models represent an hierarchical abstraction of the sructure interne of all applications deployed within the existing system.

The specification of the legacy system is based on the granularity of applications, it is represented as a functional tree. The root of the tree represents the main application (function), the nodes represent the sub-functions that implement the main function, the sub-nodes represent the sub-functions that implement functions in higher level of granularity, and the leaves of the tree represent the primitive instructions that implement functions in higher hierarchical level on the tree. Figure 5, represents an UML metamodel defines the hierarchical structure of a *Component-based Function Tree Model*. In this metamodel, we presents the hierarchy of different functions constituting an application, and the data flow (output and input data) between functions.

B. Service Components Identification

The target SOA-based system must align its business processes with environement business goals, in order to improve its ability to respond rapidly to marcket forces.

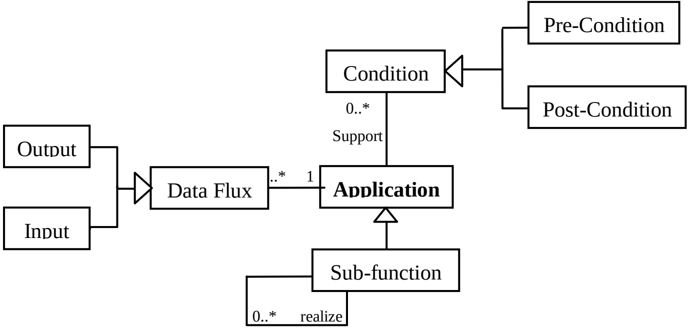


Figure 5. UML Meta-model of Component-based Function Tree Model

The first question in an SOA-based system devlopment project is how to identify business services, which responds to cunstomers' needs. This question incorporates, firstly, the problem of extraction of the relevant services from existing (legacy) system, secondly, the problem of definition of new services, which respond dynamically to the environement needs. SCDD approach for developing an SOA that we propose is a code-based approach, since the business rules that implement the core of the target enterprise system can be analyzed from the enterprise application models. In this phase, we identify the parts of condidat code that implement the business logic of future Web services. This phase consists of the *Extraction of service components*; here we extract the code for future services according to business needs of the enterprise, in order to aligning the code of existing legacy applications to the business needs discussed in the previous phases. We propose to determine which of those existing features can be exposed as Web services, to apply a method of *mapping models*. This method aims to associate the nodes of the Component-based Functional Tree model of an application components to the nodes of the model of a business process model of basic objectives.

The activities of the phase detection code that implements the web service based on the projection that we propose are summarized as follows:

- The first task is the selection of an activity of a basic business processes, which is afterwards associated to the component that contains functions performing this activity, by analyzing the final result specified for each function.
- The second task is to check if an instruction affects the final result returned by the activity of selected processes. Beginning with primitive instructions of the selected component, and rising to the nodes.
- If the selected component returns some variables required to the activity, it is necessary to select the node that is at a higher level in the model.
- else, if the variables returned are equivalent to the selected activity, the function is defined as the future public service that implements the activity of selected elementary process.
- These activities are applied to each node model targets, in order to best meet the business needs of the company.

C. *Service Components Deployment*

This step attempts encapsulate parts of candidate web services identified in the previous phase, in service components, and to defining the required interfaces (references) and interfaces provided (services) for the execution of each component or each service.

D. *Service Identification and Publication*

The final step is the technical wrapping of the functions defined and encapsulated in Service component in the preceding phases. This comprises the definition of Web Service Description Language interfaces (WSDL) that specify the Service Components interfaces. We summarize the different activities of this phase as follows:

- Identification of Service component interfaces provided for them published in the form of web services, and data requirements for their invocation.
- Definition the functionality of web services WSDL. Each Web service exposes an interface that defines the message types and patterns of trade. To do this we must first specify a well-defined interface for each service detected. Next, we turn to the definition of the functionality of web services WSDL.
- Publication services. Finally, the resulting web services must be registered in the UDDI, for use by other customers or other service-oriented systems, integrating them into a business process.

IV. Case Study and Some Implementation Aspects

In order to evaluate the SCDD approach, we take as a case study example the enterprise Production Machine Tools (PMT). PMT is a commercial enterprise, which produces and markets machine tools. For over 30 years, this enterprise uses to manage its service a monolithic inflexible information system implemented in *Basic* (Beginner's All-purpose Symbolic Instruction Code) programming language. Year by year enterprise's activities grow and the number of its clients increase, however the information system capacities remains incapable to follow this evolution while the majority of its activities are done using traditional methods through phone calls, Excel files, etc. To overcome this problem the company decides to promote its information system in order to support the new requirements as well as standards and technologies. As we know, the development of a new system costs and takes more time; so, the business directors and information system engineers decide to develop the new system on the top of the existing legacy system.

We present in this section the application of the different phases of SCDD methodology phases and some implementation details.

A. *Modeling phase*

In this phase, we analyze and model clients' business needs and existing applications in order to identify the structure of the future Service-based system. In our approach, objective and functional tree models help us to see all this laid out clearly. Our example reveals *a treatment of a client order* as a main objective.

Figure 6 shows just the beginning of an Objective Tree Model that illustrates the hierarchy of customer needs. The tree model starts with the overall objective

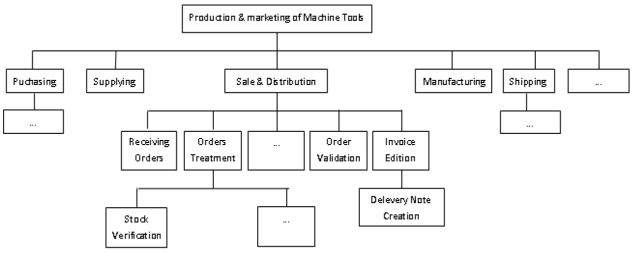


Figure 6. Goals hierarchical Tree model of the case study.

“Poduction and Marketing of Machine Tools”. The branches coming from this are sub-objectives, which are small abjectives that we need to achieve the main objective as: “*Purchasing/supplying, Manufacturing, Sale and Distribution, shipping, ... etc.*” which are necessary after to specify the business logic of the company. In reality, this tree would have many more branches and levels detailing all the sub-objectives necessary for realizing the main objective, as well as for covering all customer’s separate objectives.

B. *Service components creation*

After applying the first phase of SCDD methodology on the case study, and the extraction of the component implementing future services, we obtain finally a diagram of the hierarchy of Service Components Composing the target SOA-based application. In this section, we are describing how these components can be implemented and deployed in a distributed environment.

Figure 7 shows an example of a service component model resulting after the mapping models phase, where the *Order Client Service Component* represent a composite, which contains six sub-component.

We are choosing the JAVA programming environment ‘Eclipse’ for implementing the functionalities of the resulting component. We used the 3.5.1 version integrated already with the platform FraSCAti specification, where we affect for each component an interface; we give more detail on interface implementation in the next section.

C. *Interfaces definition and implementation*

In order to define relations between service components, different interfaces affected to each component must be developed. In this phase we create ADL descrition file in which we define the binding between different component using interfaces defined. For each component we specify the relationship with an other component it must be a relation where the component require a service from an other component as well as a relatio where the component profides an service to an other service.

V. Related Work

In this section, we present some related works, and compare the SCDD methodology with these approaches in terms of service identification strategies, SOA specification techniques as well as service deployment and programming facility.

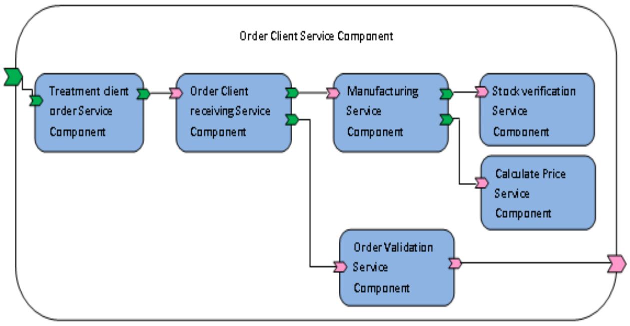

Order Client Service Component
Treatment client order Service Component
Order Client receiving Service Component
Manufacturing Service Component
Stock verification Service Component
Calculate Price Service Component
Order Validation Service Component

Figure 7. Order Client Service Component Schema.

Service identification strategies; Several approaches of SOA development are available that use different strategies to identify services, either Buttom-up ones: Chang and Kim [18], Top-down ones: Emig and al. [17], and Erl [19], or Hybrid ones: SOMA (Service Oriented Modeling and Architecture) [20], SOAD (Service-Oriented Analysis and Design) [18], CSOMA (Contextual Service Oriented Modelling and Analysis) [21], also the projects of Papazoglou and Heuvel [10]. Comparing these approaches with the SCDD approach that we present in this paper, we use in our methodology a model based strategy for identifying services from the point of view of both the producers and requesters of service, so our approach align more the business process of the target application with the client and environement needs.

Specification techniques; The existing approaches that we discussed in the second section propose to use existing modeling techniques and present a set of models or meta-models supports. For example, the approach of SOAD and SOMA that use the business oriented models (BPM: Business Process Modeling) and the object oriented models (OOAD: Object Oriented Analysis and Design) for the development of SOA, and the approach of Papazoglou and Heuvel [10], which uses the development based component (CBD: Component Based Development).

In the proposed approach, we use a business oriented service component development models in order to highlight the advantages its properties of composition and autonomy, to identify services in the right level of granularity, also reconfiguration property of component that permit to services to project easly different changes on components implementing them. The SCDD methodology view service as component entities, which can romotly eccessed independently as possible form the underlying implementation technologies.

Deployment and programming; The Service component-oriented methodology facilitates the deployment and realization of distributed service-oriented applications. The SCDD methodology, compared with existing design and development methodologies which achieve a complex architecture and not deployable in different platforms, uses service component-based models that define a well structured service-oriented application independent from technologies.

VI. Conclusion and Future Research

In this paper, we have presented the SCDD methodology considering the enterprise’s requirements as well as the existing legacy systems. SCA is a standard for distributed SOA-based systems. We motivated and described how the behavior that implements legacy system business can be aligned to enterprise needs, and modeled as service components to be used first to identify single relevant services in right granularity, and second to develop and integrate easily these services in business processes that implement the target SOA-based system.

We focused mainly in this approach on the problem of the definition of the concept *service*. Thus, we proposed in the first stage of our approach to use objective tree to model the hierarchical nature of the requirements, or business needs for designing the system. The service components model to implement and deploy the elements of the SOA considering its characteristics of reuse, autonomy and distribution. The use of this specification allows technology using service to benefit from distribution, autonomy and composition properties of service component.

Although SCDD is not the first approach that combines software engineering approach and service component models, its objective is very important, as it permits to treat the problem of SOA development starting from the modeling phase in order to describe how services constituting SOA based-application are structured. Generally, the Enterprise Service Bus (ESB) is used to implement an SOA-based system [22]. Our future research has to establish links between characteristics of SOA enterprise models based on service components and their realization in the ESB. In other words, although there exist works that permit to publicize service component as web service [23], they are still technical solutions. So, we must reveal how to realize an SOA based on service component in the ESB. This comprises the problem of monitoring and controlling, as well as the problem of adaptability and security requirements specified in enterprise models and realized in the SOA.

References

[1] F. Tonic, M. Boulier, B. Paroissin, J. Clune, F. Bernnard, and M. Gardette, “SOA : votre nouvelle architecture,” le magazine du développement : Programmez !, N° 78, June 2006.

[2] J. Zhao, M. Tanniru, and L. Zhang, “Service Computing as the Foundation of Enterprise Agility: Overview of Recent Advances and Introduction of Special Issue,” Proc. Inf Syst. LNCS, Springer Press, March 2007, vol. 9, pp. 1-8, doi : 10.1007/s10796-007-9023-x.

[3] P.M. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Service Oriented Computing: a research roadmap,” International Journal of Cooperative Information systems, vol. 17, June 2008, pp. 223-255, doi: 10.1142/S0218843008001816.

[4] L. Seinturier, Ph. Merle, R. Rouvoy, D. Romero, V. Schiavoni and J. B. Stefani, “A component-based middleware platform for reconfigurable service-oriented architectures,” International Journal Software Practice & Experience, vol. 42, May 2012, pp. 559-583 , doi: 10.1002/spe.1077.

[5] tuscany.apache.org. visited: 08.04.2013.

[6] newton.codecauldron.org . visited: 14.04.2013.

[7] frascati.ow2.org. visited: 08.04.2013.

[8] Ch. Heubès, “Mise en œuvre d’une SOA : Les clés de succès,” Xebia France, Février 2008, http://blog.xebia.fr/2007/08/16/mise-en-oeuvre-dune-soa-les-cles-du-succes/, visited : 10.04.2013.

[9] O. Zimmermann, P. Krogdahl, and C. Gee, “Elements of Service-Oriented Analysis and Design,” ftp://ftp.software.ibm.com/software/webservices/SOADpaperdWv1.pdf. visited: 15.04.2013.

[10] M. Papazoglou and P. W., Heuvel, “Service-oriented design and development methodology,” International Journal of Web Engineering and Technology (IJWET 06), vol. 2, July 2006, pp. 412-442, doi: 10.1504/IJWET.2006.010423.

[11] O. Zimmermann, N. Schlimm, G. Waller, and M. Pestel, “Analysis and Design Techniques for Service-Oriented Development and Integration,” http://ozimmer.de/download/INF05-ServiceModelingv11.pdf. visited: 15.4.2013.

[12] A. Kenzi, “Ingénierie des Systèmes Orientés Services Adaptables: Une Approche Dirigée par les Modèles,’’ PhD thesis, Ecole Normale supèrieure d’Informatique et d’Analyse de Systèmes, Université Mohamed V, Rabat. Octobre 2010.

[13] L. Seinturier, Ph. Merle, D. Fournier and N. Dolet, “Reconfigurable SCA Applications with FraSCAti Platform,” Proc. IEEE International Conference on service Computing (SCC 09), IEEE Press, June 2009, pp. 268-275, doi: 10.1109/SCC.2009.27.

[14] F. Baude, D. Caromel, C. Dalmasso, M. Danelutto, V. Getov, L. Henrio, and C. Pérez, “GCM: A grid extension to Fractal for autonomous distributed components,” Annals of Telecommunications, vol. 64, September 2009, pp. 5-24.

[15] A. Solange, A. Ludovic, B. Tomàs, C. Antonio, M. Eric, and S. Emil, “Specifying Fractal and GCM Components With UML,” Proc. IEEE International Conference of the Chilean Computer Science Society (CCS 07), IEEE Press, 2007, pp. 53-62, doi: 10.1109/SCCC.2007.17.

[16] L. Seinturier, “Le modèle de composants Fractal”, http://www.lifl.fr/~seinturi/middleware /fractal.pdf, 2008.

[17] C. Emig and S. Abeck, “Development of SOA-Based Software System and Evolutionary Programing Approach,” Proc. The Advanced International Conference On Telecommunications and on International Conference on Internet and Web Applications and Services (AICT/ICIW 06), 2006, pp. 182-187.

[18] S. H. Chang and S. D. Kim, “A Service-Oriented Analysis and Desing Approach to Dveloping Adaptable Services,” Proc. IEEE International Conference on Services Computing (SCC 07), IEEE Press, July 2007, pp. 204-211, doi: 10.1109/SCC.2007.16.

[19] T. Erl, “ Service-Oriented Architecture (SOA): Concepts, Technology, and Design,” Prentice Hall, 2005, ISBN: 0131858580.

[20] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Gariapathy, K. Holley, “SOMA: Service-Oriented Modeling and Architecture: How to identify, specify and realize services for your Service Oriented Architecture (SOA),” IBM Systems Journal, vol. 47, July 2008, pp. 377-396, doi: 10.1147/sj.473.0377.

[21] K. Boukadi, “Coopération inter-entreprises à la demande : Une approche flexible à base de services adaptables,” Thèse de doctorat de l’Ecole supérieure des Mines de Seint-Etienne, Novembre 2009.

[22] H. Christophe, “Exposer ses composants Fractal en Web service dans Petals ESB #2,” http://chamerling.org/2010/06/08/exposer-ses-composants-fractal-en-webservice-dans-petals-esb-2/, visited : 11.04.2013.

[23] Z. El Hafiane, M. Ghaoui, S. Harmach, A. Maalej, and N.M.Tourè, “Composants Fractal et Web Services,” http://deptinfo.unice.fr/twiki/pub/Minfo05/DepotDesRapports/Rapport-TER-9.pdf, visited: 15.04.2013.

Cloud Terminals for Ticketing Systems

João Ferreira, Porfírio Filipe
Instituto Superior de Engenharia de Lisboa
Lisbon, Portugal
{jferreira, pfilipe}@deetc.isel.pt

Gonçalo Cunha, João Silva
Link Consulting, SA
Lisbon, Portugal
{goncalo.cunha, joao.r.silva}@link.pt

***Abstract*— In this research work, we introduce the concept of a thin device implemented on a cloud platform for terminal devices on the front end of ticketing systems. Therefore, we propose the evolution of the traditional architecture of ticketing for a cloud based architecture in which the core processes of ticketing are offered through a Software-as-a-Service (SaaS) business model, which can be subscribed by transport operators that pay-per-use. Ticketing terminal devices (e.g., gates, validators, vending machines) are integrated in the cloud environment creating the concept for a 'thin' device. This approach is achieved by moving business logic from terminals to the cloud. Each terminal is registered to be managed by each own operator, configuring a multi-tenancy implementation which is vendor hardware independent, allowing to address elasticity and interoperability issues. The elasticity of the cloud will support the expansion/implosion of small (transport) operators business around electronic ticketing. In the near future, this ticketing solution will promote collaboration between operators.**

Keywords-Cloud Computing; Software as a Service (SaaS); Ticketing System; Terminal Device;

I. INTRODUCTION

In this work, we propose the definition of a thin device terminal in an Android platform that allows building a common transportation ticketing services to which the terminals can connect through a simple Plug-and-Play model that reduces human interventions.

Therefore, it will be necessary to define the architecture of the cloud services, as well as the characteristics of terminals to consume those services. The goal is to achieve global consolidation of all business logic and move terminal specific logic to the cloud; therefore reducing the overall system complexity.

This change of paradigm benefits from the fact that cloud ticketing services can be accessed through the Internet and they can be elastically grown or shrunk, providing easier scalability and high availability.

In this paradigm, the consolidation of business logic facilitates the adoption of open and secure protocols, making the terminal simple benefiting from being online with the global ticketing system to offer value-added features on lower capacity terminals.

In the aviation industry, there are already systems for seat reservations and ticketing to be offered "as a service" for several airlines, often at a cost of only pennies per ticket [1]. In fact, very few low cost operators manage and maintain its own ticketing system because SaaS options available in the market do it more efficiently and at a lower cost [2, 3].

There are several advantages in having lightweight terminals connected to cloud business logic, such as: (1) consolidated logic with easier maintenance and lower IT costs; (2) improved physical security (avoid secure elements distribution and logistics); (3) enable functionality by subscription for devices; (4) support offline and online operation models over the same infrastructure; and (5) reduced complexity for supporting new terminals, by using open interoperable protocols.

This paper is organized in seven sections: Section I defines the work context; Section II describes the electronic ticketing survey; Section III describes the proposed cloud ticketing architecture; Section IV describes the proposed thin device concept; Section V describes the proposed multitenant approach; Section VI describes the device provisioning associated with thin device implementation; and finally, Section VII presents the conclusion.

II. ELECTRONIC TICKETING SURVEY

An electronic ticketing system is designed to enable fare collection in a simple, secure and efficient way. In the public transport operators market, electronic ticketing is associated with a trip or a set of trips using transportation service. A survey of electronic ticketing systems can be found at [4, 5].

The customer obtains an electronic ticket medium (smart card, mobile device ticket) which is the storage location for electronic tickets. When an operator sells a ticket, the sale is registered in the ticket storage medium and will be validated before use [6]. In association with the sale process of electronic tickets, electronic information is stored and processed for the purpose of producing: (1) Billing data are used in the sharing of ticket revenues among the various operators involved in the ticketing system, (2) Revenue data, and (3) Statistics (about the sale and use of tickets).

An electronic ticketing system may also incorporate a number of other functions: (1) Ticket inspection function; (2) Internet services (for example online sales of tickets); (3) ticket vending machine; and (4) entrance/exit ticket validation machines. This operation is performed at front-end system (entrance and exit ports) with dedicated equipment and private network.

A ticketing system operation is based on a token issuing entity (issuer), a set of users, tokens, and verifiers who verify whether tokens are valid, performed in a dedicated solution using a private network to deal with security and privacy

ISBN: 978-1-61208-270-7

issues. Typically, a user U must buy a token from token issuer I. Therefore, U selects his desired ticket and pays it. Issuer I then checks whether U is eligible to obtain a token (e.g., whether U paid for the ticket), and, if applicable, issues a token T and passes it to U. From now on, U is able to use token T to prove that he is authorized to use the transport network. This means that every user who is in possession of a token that has been issued by a genuine issuer is considered an authorized user. For instance, a user U wants to travel from a place X to some location Y. Before U is allowed to enter the transport system at X, he must first prove to a verifier Vin, at the entrance of the transport network that he is authorized to access it. When Vin verifies successfully the user’s token, U is allowed to enter. Otherwise, access will be denied. On the other hand, during his trip, U may encounter arbitrary inspections where he must prove that he is authorized to use the transport network. Thus, a verifier V may check the user’s token T. If verification of T is successful, U is allowed to continue his trip. Otherwise, U must leave the transport network and may be punished for using it without authorization. After arriving at location Y, the user’s token T can be checked for a last time. Again, if T cannot be verified successfully, U may be punished.

Note that a token is typically bound to some limitations. For instance, it may be bound to some geographical or time usage restrictions. Additionally, a token may be bound to the identity of its owner (i.e., the entity that bought the ticket).

Most of these ticketing systems are based on proprietary solution with terminal at transportation stop and a central system to handle all related operations.

A. A Review of Proprietary Solutions

In Europe, there exist several implementations of the e-ticketing paradigm, mainly on the national level (limited to a single country). The information concerning system specification is for the most part publicly unavailable, which is a hurdle when a review of privacy solutions in the area is considered. However, certain pieces of information are openly accessible. Main European platforms are: (1) In the UK, ITSO (Integrated Transport Smartcard Organization) has developed a specification for interoperable smart ticketing [7], which is similar to the guidelines of the respective standards; (2) Another popular proprietary e-ticketing standard developed in Europe is called "Calypso" [8]; and (3) The e-ticketing systems based on MIFARE cards, such as Dutch OV-chipkaart, London's.

B. Main Ticketing Project

One of first initiatives in electronic ticketing was the Cubic Transportation Systems [9]. From this project, several world projects emerged. Some of them are already operational in countries like United Kingdom, Germany, France, Australia, Netherlands, and South Korea.

One of the first projects that used the contactless smart card based ticketing was Octopus [10]. It started in 1994, and became operational in the year 1997 in Hong Kong. The system was built by AES Prodata [11], which is a member of ERG Group [12] using the Sony Felica card [13] for contactless payments. The company ERG Group owners automated fare collection systems cooperating also with transport system projects in Manchester and Hertfordshire (Great Britain). The name of the project is Herts Smart Scheme using the Philips [14] platform for ticketing MIFARE. It is based on the EPT policy, is having different cards to handle special fares and is operative since 1997.

Another ERG Group project is Metrebus Card [15] operative at the moment in Roma (Italy). Metrebus Card uses a combi-card, which stores tickets like the project SIROCCO in Spain but it does not have an anonymous option.

In San Francisco (USA) ERG group has implemented the project TransLink [16] that will start its first phase in 2003. In this project, ERG Group works with Cubic Transportation Systems to develop an EPT solution which uses in the beginning a personalized card.

Another project that started around 1995 was ICARE [17]. It concerns Lisbon (Portugal), Constance (Germany), Venice (Italy) and Paris (France). This project evolved, with the entrance of Brussels into the consortium, creating a telematics platform, which defines a card-terminal ticketing standard called Calypso [18]. The protocol used in ICARE was a proprietary protocol developed by Innovatron and further on implemented by ASK [19] in France.

III. CLOUD TICKETING ARCHITECTURE

The vision of the present proposal is illustrated in Figure 1, where a set of dedicated services are available in an SaaS approach and front end devices (e.g., validators, vending machines, gates and others) ‘migrate’ from an integrated fat device to a flexible and modular thin device with all or part of business process logic executed remote in a SaaS approach. The idea is to interact with several equipment interfaces and integrate related business process in a SaaS approach.

The proposed idea for a ticketing system on the cloud can be simply described as a set of standards based services on the cloud to perform a specific business function (e.g., card issuing, ticket sale). These services are available through a communication protocol that is common to all registered devices. When front office devices (PCs, POS, Smartphones, tablets, web browsers, etc) first announce themselves to the cloud services, they identify themselves, as well as the tenant they belong to and automatically downloading the relevant software and configurations for the functions assigned to them. After the registration occurs, the device is able to interact with the cloud services, for instance a tablet computer connects to the cloud provisioning service, authenticates itself, and automatically downloads the ticket sale software. Afterwards, is allowed to start selling tickets to customers.

The proposed architecture is composed of the following layers of services (see Figure 1):

- Data Access Services – internal services to access business data (customers, cards, sales, validations, etc);
- Business Services – cloud exposed services to implement business operations like registering a new customer, authorizing a ticket sale for a specific customer, or consulting a catalog of tickets available to the specific card;

• Business Process Services – services that coordinate among multiple business services to implement specific use cases, e.g., ticket sale use case, which generally involves: (1) read the card; (2) browse the ticket catalog for available products; (3) choose the ticket to buy; (4) pay; (5) load the card; and (6) confirm and register the sale. The output of this service is the information to present to the user on the screen, as well as available operations. The inputs of the service are the actions performed by the user.

The Data Access Services Layer is a lower level internal layer, used to abstract the access to the data provider.

The Business Services Layer should implement the service business logic of the overall system, including data validations, user authorization, accounting algorithms and data correlation.

Here, we highlight the proposed cloud services on the Business Services Layer: (1) Customer Service; (2) Card Service; (3) Ticket Sale Service; (4) Validation and Inspection Service; (5) Device Provisioning Service; and (6) Ticket Catalog Service.

In order to implement a full ticketing system multiple use cases must be considered. However, we highlight only the relevant use cases, which are included on the Process Coordination Services Layer: (1) Ticket Sale Business Process Service; (2) Customer Registration Business Process Service; (3) Card Renewal Business Process Service; and (4) Card Cancellation Business Process Service.

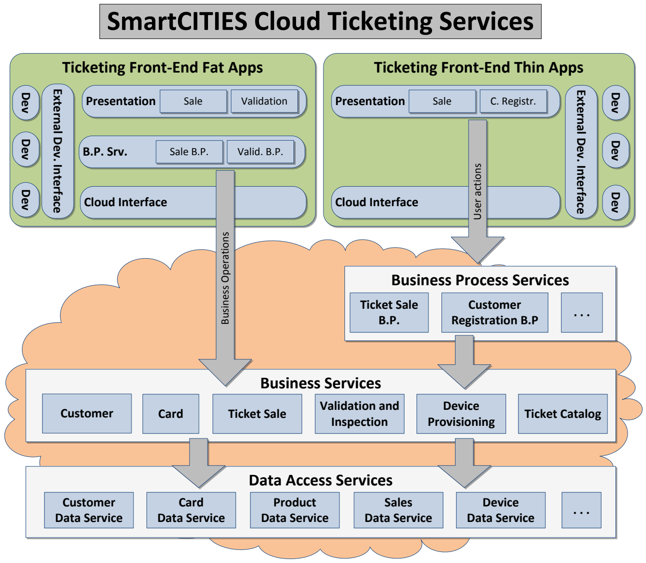


Figure 1. Cloud Ticketing Architecture.

To complement the exposed cloud services, there is also a range of back office applications, to manage the system as a whole (e.g., Customer relationship management, Product Catalog Management, Reporting, Device Management, etc).

The interoperability goal implies the existence of common security and privacy measures (e.g., an agreement on mutually recognized and accepted security and privacy suits).

The need for security is widely acknowledged by transport companies, since insecure solutions may result in substantial revenue losses.

Privacy, namely customer privacy, to the contrary, is not in direct interest of service providers. The reason for this is that possible risks associated with privacy violation have far less serious implications for company business compared to security. The interoperability goal poses a further challenge to privacy since sharing of privacy-critical data, which is needed for a proper delivery of transport services by cooperating companies, should be performed in a privacy-preserving way.

A. Cloud Ticketing vs Traditional Ticketing

In the cloud, ticketing system architecture is based on consuming services organized in a layer available in a cloud platform. The services have published interfaces. These interfaces support the development of personalized ticketing systems. The main effort is the definition and development around the implementation of services. Cloud platform is important to reduce/manage hardware costs and to publish the transport operator’s services. Therefore, the cloud ticketing has the following benefits:

- Reduce system development cost: the creation of a robust service layer available in a cloud platform has the benefit of a better return on the investment made in the creation of the software. Services map to distinct business domains, opening the possibility of personalized ticketing systems for small transportation companies, with small budgets.
- Code mobility: since location transparency is a property of a service-oriented architecture, code mobility becomes a reality. The lookup and dynamic binding to a service means that the client does not care where the service is located. Therefore, an organization has the flexibility to move services to different machines, or to move a service to an external provider.
- Focused developer roles: cloud ticketing approach will force an application to have multiple layers. Each layer has a set of specific roles for developers. For instance, the service layer needs developers that have experience in data models, business logic, persistence mechanisms, transaction control, etc.
- Better testing/fewer defects: services have published interfaces [20] that can be tested easily writing unit tests.
- Support for multiple client types (multitenant, see Section V): as a benefit of a service-oriented architecture on a cloud platform, personalized ticketing systems can be easily developed.
- Service assembly: the services will evolve into a catalog of reusable services. Everyone will benefit from new applications being developed more quickly as a result of this catalog of reusable services. The big issue on this topic is the standardization.
- Better maintainability: software archeology [21] is the task of locating and correcting defects in code. By focusing on the service layer as the location for your

business logic, maintainability increases because developers can more easily locate and correct defects.

- More reuse: code reuse has been the most talked about form of reuse over the last four decades of software development.
- Better parallelism in development: the benefit of multiple layers means that multiple developers can work on those layers independently. Developers should create interface contracts at the start of a project and be able to create their parts independently of one another.
- Better scalability: one of the requirements of a service-oriented architecture is location transparency. Typically, to achieve location transparency, applications lookup services (in a directory) and bind to them dynamically at runtime. This feature promotes scalability since a load balancer may forward requests to multiple service instances without the knowledge of the service client.
- Higher availability: because of location transparency, multiple servers may have multiple instances of a running service. When a network segment or a machine goes down, a dispatcher can redirect requests to another service without the client's aware.
- Main disadvantages: the security and the privacy of data and latency issues.

IV. THIN DEVICE CONCEPT

An e-ticketing system manages terminal devices to sell and validate (check passengers authorization) tickets before, during and after the travel. This system provides an alternative to conventional ways for proving the existence and validity of the travel rights (e.g., paper tickets) through transferring the needed information to an electronic storage medium (e.g., an RFID card). For more details about radio frequency identification (RFID) see [22].

An e-ticketing system can be coarsely analyzed into two main parts: front-end devices and back-end systems. A front-end device is a terminal (described in Figure 2) with a RFID card reader that interfaces with the back-end systems. Typically, these terminals include business logic to control their functionalities, mainly for selling or validating tickets). These devices communicate to an IT infrastructure to request information and report performed transactions. This kind of terminal device, in current electronic ticketing systems, we designated by fat device.

Our main idea of the current work is illustrated in Figure 3, where we propose: a common interface to card readers (and other peripherals); and a common interface to the devices where the business logic is located in the cloud. The adoption of the thin device (client) concept opens several issues, such as dependency of communications, high latency issues (usually, a validation process at a gate should take less than 300ms) and additional security and privacy issues.

The main advantage of the thin device adoption is the easier implementation of new devices and the reuse of business logic across multiples types of devices, facilitating software updates. Figure 4 shows this concept, where different terminals devices can share common modules and a dedicated company terminal device can be created by the change business process, presentation layer and perhaps the card reader process (only in the case of the usage of a different smart card).

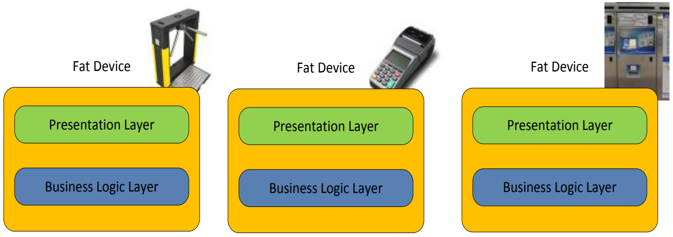


Figure 2. Fat Device architecture.

As in the case of a thin device, the term is often used to refer to software, but it is also used to describe the networked computer itself. When the applications require multimedia features or intensive bandwidth it is important to consider going with thin devices/clients. One of the biggest advantages of fat clients rests in the nature of some operating systems and software being unable to run on thin clients. Fat clients can handle these as they have their own resources.

The proposed architecture for cloud ticketing systems is designed to support two sets of front-end devices on the customer side: the ones with lower processing capacity but are always online; the other that at some point in time need to work online but have higher processing capacity. The first set of devices has what we called "thin apps", the second set are the "fat apps".

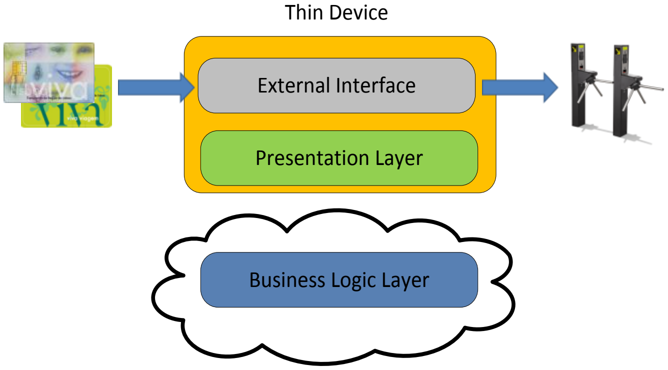


Figure 3. Thin Device architecture.

Thin apps know few about business logic and have presentation logic built-in. Typically, they receive the screens to be displayed and send back requests. Global process coordination and business logic are located in the cloud. The operation depends on network connection to access cloud services on the Business Process Services Layer. In Figure 5, we show a generic workflow of a thin app performing an action in a cloud service, where a few points are highlighted, namely:

• The thin app interacts with one business process service, which coordinates multiple business services;

• The thin app receives presentation information and sends back commands;

• Every app interaction communicates with the cloud;

• When the operation ends, the app sends an action which generates a confirm operation (e.g., ticket sale confirmation). The confirm operation commits the information to persistent storage.

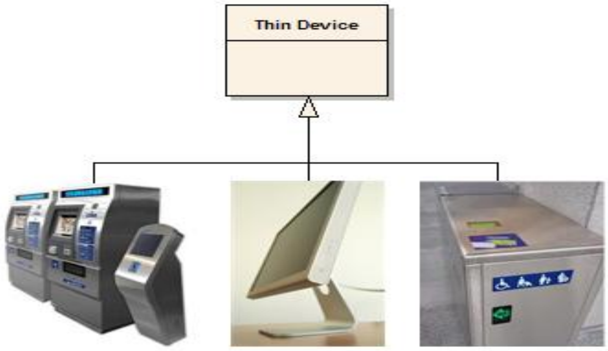


Figure 4. Usage of thin device approach.

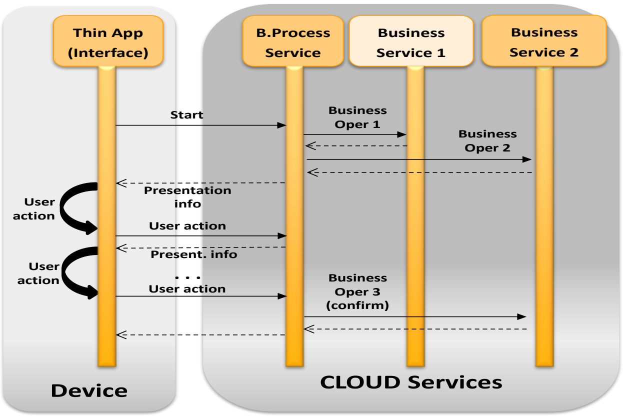


Figure 5. Thin client workflow.

On the other hand, fat apps are, for instance, running in PCs, tablets or even POS and typically have the process coordination installed locally and some offline data to enable offline usage. In ticketing applications, they are still required for some use cases, where short timing requirements exist and offline capability is a must. An example is a ticket validation device aboard a bus. In the bus scenario, there are zones of the route without network coverage and the timing requirement from the moment the user puts the cards on the validator till the moment of the approval should be bellow 300ms [23].

Figure 6 shows a generic workflow of fat apps, the general case is to have the process coordination installed in the device, with local interactions (e.g., ticket validation), and at the end of the operation the cloud service is informed of the operation result. A generic workflow of the interactions follows the highlights:

• The fat app performs every interaction locally (possibly using cached reference data);

• The fat app periodically sends the confirm operations to the cloud service (e.g., ticket validations). These confirm operations commit the information to persistent storage

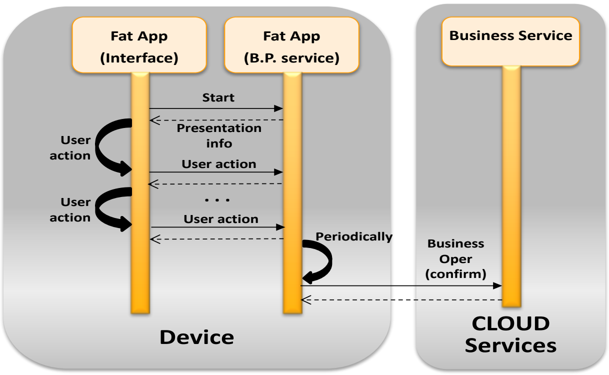


Figure 6. Fat client workflow.

A. Thin Device Development

Our development is based on an Android device with the following characteristics: (1) usb host; size bigger than 4,1’’; communication 3G or 4G; android version 4.0 or upper; NFC (near field communication), two USB with dedicated power. Our first development step was the communication interface between the reader (terminal) of RFID chip with the standard ISO14443 [24], which consists of: (1) physical characteristics; (2) radio frequency interface power and signal interface; (3) initialization and anti-collision; and (4) transmission protocol.

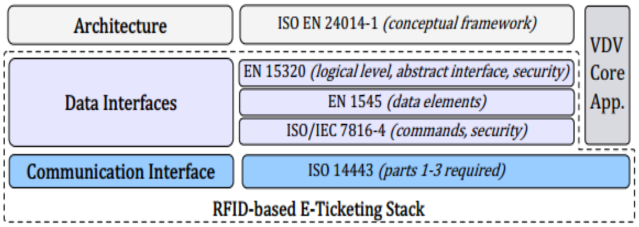

Architecture
ISO EN 24014-1 (conceptual framework)
Data Interfaces
EN 15320 (logical level, abstract interface, security)
EN 1545 (data elements)
ISO/IEC 7816-4 (commands, security)
VDV Core App.
Communication Interface
ISO 14443 (parts 1-3 required)
RFID-based E-Ticketing Stack

Figure 7. Standards to implement thin device concept.

We transfer most of adopted standards available in Calypso e-ticketing system [25] adopted in several countries such as, Belgium, Canada, China, France, Israel, Italy, Portugal, etc.). Figure 7 illustrates our standards for this interoperation necessary for thin device concept implementation.

The standard, ISO EN 24014-1 [26] introduces a conceptual framework for developing an interoperable architecture for transport fare management systems. It describes the structure of an interoperable platform, its main actors, and general flows of information exchange. Privacy is considered at a conceptual level by requiring the definition of a security scheme that should provide privacy.

The data interface layer, EN 15320, defines the logic data structure in the card and defines the communication interface between the card and the terminal. The card data interface and data group interface handle security based in the specification of the security subsystem (SSS), illustrated in Figure 8. Security related operations are defined in the card profile and data group profiles (this to handle privacy issues).

The ISO/IEC 7816-4 defines the commands exchange as well as the retrieval of data structures and data objects in the cards. Security is taken into account specifying the methods for secure messaging and security architecture defines access rights to files and data in the card. A list of algorithms is available in [27].

Communication interface layer is based on ISO 14443 [28], which handles the connection of terminal to the card reader. The communication between terminal and the cloud is based on our work described by [2].

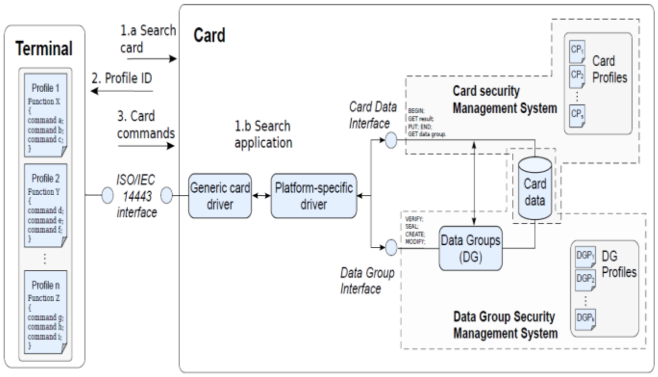


Figure 8. Interaction process between card and terminal.

V. MULTI-TENANCY

Multi-tenancy, which lets multiple tenants (users) share a single application instance securely, is a key enabler for building such a middleware. The main idea is to apply this concept to terminals devices adopting the thin device concept to consolidate globally the available operations in ticketing back offices systems. Multi-tenancy has been proposed as a way to achieve better resource sharing and to provide almost zero cost when unused.

In order to support multi-tenancy on the cloud services it is important to consider that multiple operators may be organized in a common metropolitan area, having some (not all) common customers, smartcards and multi-modal tickets [2]. In these cases, it is important to have a consolidated view of common business information (customer, cards, sales, validations, etc.) by all operators to enable revenue distribution.

With this target scenario, we propose a hierarchy of tenants with multiple roots (see Figure 9). Each root is a transport area with multiple operators where some parts of the business information (customers, cards, etc) are common to several operators.

The hierarchy of tenants has the following rules:

• Lower level tenants (operators) can view information about their private customers, as well as business information common to the metropolitan area.

• Upper level tenants can read and consolidate common business information to the lower level tenants (e.g., customers, cards, sales and validations).

• Upper level tenant may not see information about private customers, and sales/validations of private tickets.

Here, we discuss the option of having a shared database or separate database/schema implementation of multi-tenancy.

The main concerns with this decision were privacy, security and extensibility. It is necessary to avoid risks of having one operator accessing information belonging to other operators (they may be competitors). On the other hand, it is very common for an operator to require customizations specific to its business. Therefore, we have chosen to have a separate database approach.

With the requirement of having a hierarchy of tenants, using a separate database approach, has an additional challenge – how to consolidate common business information (e.g., sales of multi-modal tickets) on the upper levels, which is generated at the lower levels?

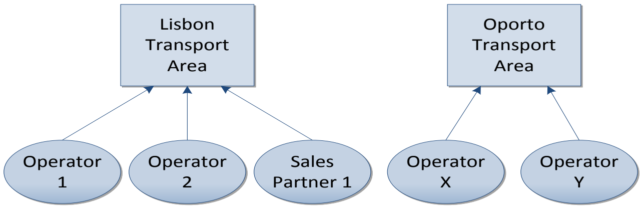


Figure 9. Sample hierarchical structure of tenants.

The answer is to have the lower levels ship the common business information to the upper levels, where it is consolidated and becomes its master repository. Private information on the lower levels is never shipped.

VI. DEVICE PROVISIONING

To enable device installation automation, we propose a device provisioning model, where devices can detect and download the relevant software to support the functionalities assigned to them. This procedure depends on cloud platform and device operating system. This is a working project, where we are starting our approach with a Windows Azure platform and an Android operating system.

In this model, we assume that devices have some generic bootstrap software pre-installed, that asks the cloud service to provide it with the device specific software it needs. Either this bootstrap could be factory installed or user installed, however it does not bring any device specific configurations. When the bootstrap starts (Figure 10), it sends a bootstrap command to the cloud, requesting information on what it needs to install, and receives back information about the download location and metadata. Next, it downloads the software and configurations and starts the newly installed software.

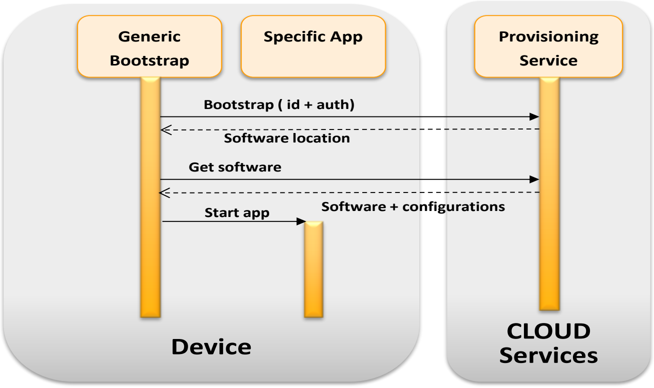


Figure 10. Provisioning workflow.

VII. CONCLUSION

The proposed system uses a novel approach based on SaaS model for the development of a personalized ticketing software. This project started 12 months ago in a synergy between the technology company Link Consulting [29], with 10 years of experience in ticketing business and researchers in computer science at ISEL [30]. Starting from the initial kickoff meeting several Masters Thesis are running: (1) network security, where we study the privacy and security problems of moving business logic from terminals to the cloud. This topic not covered in this paper handles the problems of losing connectivity and the security issues; (2) Cloud Computing projects, where we are developing the concept for different cloud platforms as well the implementation of a set of services regarding the complete ticketing process.

The architecture described in this paper illustrates the adoption of the thin device concept within our cloud ticketing approach. In current work stage we have an Android platform, where we 'migrate' current process of a selling ticketing operation and validation process at a gate using calypso platform [2]. Part of this research effort belongs to Link company in the scope of the "SmartCITIES Cloud Ticketing" project [28], which is focused on designing an interoperable, cost-efficient, multi-supplier, cloud based ticketing solution, where transport operators may opt in and out when they need/want to.

This project brings together two complementary sets of experiences: the engineering experience applied to ticketing solutions of Link Consulting [28] and the computer science and research experience of ISEL – Instituto Superior de Engenharia de Lisboa, which lays out the path to a solid foundation of a cloud ticketing solution.

REFERENCES

[1] "Impact of changes in the airline ticket distribution industry," www.gao.gov/assets/240/239237.pdf, [retrieved: April, 2013].

[2] J. C. Ferreira, P. Filipe, C. Gomes, G. Cunha, and J. Silva, "Taas – ticketing as a service," in proc. of CLOSER 2013 - 3rd Int. Conf. on Cloud Computing, 8-10 May, Aachen-Germany.

[3] A. Benlian, T. Hess, and P. Buxmann, "Drivers of SaaS-adoption–an empirical study of different application types," in Business & Information Systems Engineering 1.5, 2009, pp. 357-369.

[4] M. Mut-Puigserver, M. M. Payeras-Capellà, and J. L. Ferrer-Gomila, A. Vives-Guasch, and J. Castellà-Roca, "A survey of electronic ticketing applied to transport," in Computers & Security, Volume 31, Issue 8, November 2012, pp 925-939.

[5] E.A. V. Vilanova, R. Endsuleit , J. Calmet, and I. Bericht, "State of the art in electronic ticketing," Universität Karlsruhe, Fakultät für Informatik.

[6] TCRP Report 115, "Smartcard interoperability issues for the transit industry," in Transit Cooperative Research Program, www.nap.edu/openbook.php?record_id=14012, [retrieved: April, 2013].

[7] ITSO Technical Specification 1000, "Interoperable public transport ticketing using contactless smart customer media. Version V2.1.4," http://www.itso.org.uk//Home/Itso-Specification, 2010, [retrieved: April, 2013].

[8] Frederic Levy, "Calypso functional presentation. SAM and key management," www.calypsostandard.net/index.php?option=com, [retrieved: April, 2013].

[9] "CubicCoorporation", www.cubic.com, [retrieved: April, 2013].

[10] Octopus Cards Ltd, "The Octopus Project, 1994," www.octopuscards.com/e index.html, [retrieved: April, 2013].

[11] "An overview over ticketing projects," http://www.tick-et-portal.de/, [retrieved: April, 2013].

[12] ERG Group, www.erggroup.com, [retrieved: April, 2013].

[13] "Sony felica card," http://www.sony.co.jp/en/products/felica/contents02.html, [retrieved: April, 2013].

[14] "Philips semiconductors," www.semiconductors.philips.com, [retrieved: April, 2013].

[15] "Metrebus card," http://europeforvisitors.com/rome/transportation/rome-metrebus-tickets-and-fares.htm, [retrieved: April, 2013].

[16] "TransLink," www.translink.co.uk/, [retrieved: April, 2013].

[17] "Régie autonome des transport parisiens," France, ICARE, 1998. Philippe Vappereau, Project Coordinator. http://www.cordis.lu/telematics/taptransport/research/projects/icare.html, [retrieved: April, 2013].

[18] "Calypso standard for card terminal ticketing," http://www.calypso.tm.fr, [retrieved: April, 2013].

[19] "ASK," France. http://www.ask.fr, [retrieved: April, 2013].

[20] M. Fowler, "Public versus published interfaces," in IEEE Software, Vol. 19, No. 2, March/April 2002, pp. 18-19.

[21] A. Hunt and D. Thomas, "Software archaeology," in IEEE Software, Vol. 19, No. 2, March/April 2002, pp. 20-22.

[22] Smart Card Alliance, "Transit and contactless financial payments: new opportunities for collaboration and convergence," http://www.smartcardalliance.org/resources/pdf/Transit_Financial_Linkages_WP_102006.pdf, [retrieved: April, 2013].

[23] "RFID," http://en.wikipedia.org/wiki/Radio-frequency_identification, [retrieved: April, 2013].

[24] ISO 14443 Standards family, "Identication cards - contactless integrated circuit cards - Proximity cards," www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39693, [retrieved: April, 2013].

[25] ISO/IEC 7816-4:2005, "Identication cards - integrated circuit cards - part 4: organization, security and commands for interchange," - http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=36134, [retrieved: April, 2013].

[26] Calypso Networks Association, "Calypso handbook," http://www.calypsonet-asso.org/downloads/100324-CalypsoHandbook-11.pdf, [retrieved: April, 2013].

[27] GlobalPlatform's Value Proposition for the Public Transportation Industry, "Seamless, secure travel throughout multiple transportation networks," http://www.globalplatform.org/documents/whitepapers/GP_Value_Proposition_for_Public_Transportation_whitepaper.pdf, [retrieved: April, 2013].

[28] "ISO14443," www.openpcd.org/ISO14443, [retrieved: April, 2013].

[29] "SMARTCITIES projet", http://www.link.pt/smartcities, [retrieved: April, 2013].

[30] "Polytechnic Institute in Lisbon", ISEL – www.isel.pt, [retrieved: April, 2013].

Query Optimization in Cooperation with an Ontological Reasoning Service

Hui Shi
School of Computer and Information
Hefei University of Technology
Hefei, China
hshi@cs.odu.edu

Kurt Maly,Steven Zeil
Department of Computer Science
Old Dominion University
Norfolk, USA
{maly,zeil}@cs.odu.edu

***Abstract*—Interposing a backward chaining reasoner between a knowledge base and a query manager yields an architecture that can support reasoning in the face of frequent changes. However, such an interposition of the reasoning introduces uncertainty regarding the size and effort measurements typically exploited during query optimization. This paper presents an algorithm for dynamic query optimization in such an architecture. Experimental results confirming its effectiveness are presented.**

Keywords-semantic web; ontology; reasoning; query optimization; backward chaining

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?” 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 judgments. 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. We believe that qualitative descriptors such as “groundbreaking research in data mining” are likely to be accepted as meaningful if they represent a consensus of an appropriate subset of the community at large. Once accepted as meaningful, these descriptors can be realized in a system and made available for use by all members of that community.

The system implied by these queries is an example of a semantic web service where the underlying knowledgebase covers linked data about science research that are being harvested from the Web and are supplemented and edited by community members. The query examples given above also imply that the system not only supports querying of facts but also rules and reasoning as a mechanism for answering queries.

A key issue in such a semantic web service is the efficiency of reasoning in the face of large scale and frequent change. Here, scaling refers to the need to accommodate the substantial corpus of information about researchers, their projects and their publications, and change refers to the dynamic nature of the knowledgebase, which would be updated continuously.

In semantic webs, knowledge is formally represented by an ontology as a set of concepts within a domain, and the relationships between pairs of concepts. The ontology is used to model a domain, to instantiate entities, and to support reasoning about entities. Common methods for implementing reasoning over ontologies are based on First Order Logic, which allows one to define rules over the ontology. There are two basic inference methods commonly used in first order logic: forward chaining and backward chaining [1].

A question/answer system over a semantic web may experience changes frequently. These changes may be to the ontology, to the rule set or to the instances harvested from the web or other data sources. For the examples discussed in our opening paragraph, such changes could occur hundreds of times a day. Forward chaining is an example of data-driven reasoning, which starts with the known data in the knowledgebase and applies modus ponens in the forward direction, deriving and adding new consequences until no more inferences can be made. Backward chaining is an example of goal-driven reasoning, which starts with goals from the consequents, matching the goals to the antecedents to find the data that satisfies the consequents. As a general rule forward chaining is a good method for a static knowledgebase and backward chaining is good for the more dynamic cases.

The authors have been exploring the use of backward chaining as a reasoning mechanism supportive of frequent changes in large knowledge bases. Queries may be composed of mixtures of clauses answerable directly by access to the knowledge base or indirectly via reasoning applied to that base. The interposition of the reasoning introduces uncertainty regarding the size and effort associated with resolving individual clauses in a query. Such uncertainty poses a challenge in query optimization, which typically relies upon the accuracy of these estimates. In this paper, we describe an approach to dynamic optimization that is effective in the presence of such uncertainty.

In section II, we provide background material on the semantic web, reasoning, and database querying. Section 3 formally gives the overall algorithm for answering a query. The details of the optimization methods we have developed within the backward chaining algorithm will be described in a later paper. We have implemented this algorithm and performed experiments with data sets ranging from 1 million to 6 million facts. In section 4 we report on some of these experiments, comparing our new algorithm with a commonly used backward chaining algorithm JENA [2].

II. Related Work

A number of projects (e.g., Libra [3, 4], Cimple [5], and Arnetminer [6]) have built systems to capture limited aspects of community knowledge and to respond to semantic queries. However, all of them 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 involving reasoning.

Many knowledge bases [7-10] organize information using ontologies. Ontologies can model real world situations, can incorporate semantics which 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.

Two common methods of reasoning over the knowledge base using first order logic are forward chaining and backward chaining [1]. Forward chaining is an example of data-driven reasoning, which starts with the known data and applies modus ponens in the forward direction, deriving and adding new consequences until no more inferences can be made. Backward chaining is an example of goal-driven reasoning, which starts with goals from the consequents matching the goals to the antecedents to find the data that satisfies the consequents. Materialization and query-rewriting are 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 computation of answers with data that are relatively static. Owlim [11] and Oracle 11g [12], for example implement materialization. Query-rewriting and backward chaining are suitable for efficient computation of answers with data that are dynamic and infrequent queries. Virtuoso [13], for example, implements a mixture of forward-chaining and backward-chaining. Jena [2] supports three ways of inferencing: forward-chaining, limited backward-chaining and a hybrid of these two methods.

In conventional database management systems, query optimization [14] is a function to examine multiple query plans and selecting one that optimizes the time to answer a query. Query optimization can be static or dynamic. In the Semantic Web, query optimization techniques for the common query language, SPARQL [15, 16], rely on a variety of techniques for estimating the cost of query components, including selectivity estimations [17], graph optimization [18], and cost models [19]. These techniques assume a fully materialized knowledge base.

Benchmarks evaluate and compare the performances of different reasoning systems. The Lehigh University Benchmark (LUBM) [20] is a widely used benchmark for evaluation of Semantic Web repositories with different reasoning capabilities and storage mechanisms. LUBM includes an ontology for university domain, scalable synthetic OWL data, and fourteen queries. The University Ontology Benchmark (UOBM) [21] extends the LUBM benchmark in terms of inference and scalability testing. Both LUBM and UOBM have been widely applied to the state of the art reasoning systems to show the performance regarding different aspects [20, 21].

III. Dynamic Query Optimization with an Interposed Reasoner

A query is typically posed as the conjunction of a number of clauses. The order of application of these clauses is irrelevant to the logic of the query but can be critical to performance.

In a traditional data base, each clause may denote a distinct probe of the data base contents. Easily accessible information about the anticipated size and other characteristics of such probes can be used to facilitate query optimization.

The interposition of a reasoner between the query handler and the underlying knowledge base means that not all clauses will be resolved by direct access to the knowledge base. Some will be handed off to the reasoner, and the size and other characteristics of the responses to such clauses cannot be easily predicted in advance, partly because of the expense of applying the reasoner and partly because that expense depends upon the bindings derived from clauses already applied. If the reasoner is associated with an ontology, however, it may be possible to relieve this problem by exploiting knowledge about the data types introduced in the ontology..

In this section, we describe an algorithm for resolving such queries using dynamic optimization based, in part, upon summary information associated with the ontology. In this algorithm, we exploit two key ideas: 1) a greedy ordering of the proofs of the individual clauses according to estimated sizes anticipated for the proof results, and 2) deferring joins of results from individual clauses where such joins are likely to result in excessive combinatorial growth of the intermediate solution.

We begin with the definitions of the fundamental data types that we will be manipulating. Then we discuss the algorithm for answering a query. A running example is provided to make the process more understandable.

We model the knowledge base as a collection of triples. A triple is a 3-tuple (x,p,y) where x, p, and y are URIs or constants and where p is generally interpreted as the identifier of a property or predicate relating x and y. For example, a knowledge base might contains triples

(Jones, majorsIn, CS), (Smith, majorsIn, CS),
(Doe, majorsIn, Math), (Jones, registeredIn, Calculus1),
(Doe, registeredIn, Calculus1).

A QueryPattern is a triple in which any of the three components can be occupied by references to one of a pool of entities considered to be variables. In our examples, we will denote variables with a leading '?'. For example, a query pattern denoting the idea "Which students are registered in Calculus1?" could be shown as

(?Student,registeredIn,Calculus1).

A query is a request for information about the contents of the knowledge base. The input to a query is modeled as a sequence of QueryPatterns. For example, a query "What are the majors of students registered in Calculus1?" could be represented as the sequence of two query patterns

[(?Student,registeredIn,Calculus1),
(?Student, majorsIn, ?Major)].

The output from a query will be a QueryResponse. A QueryResponse is a set of functions mapping variables to values in which all elements (functions) in the set share a common domain (i.e., map the same variables onto values). Mappings from the same variables to values can be also referred to as variable bindings. For example, the QueryResponse of query pattern (?Student, majorsIn, ?Major) could be the set

{{?Student => Jones, ?Major=>CS},
{?Student => Smith, ?Major=>CS },
{?Student => Doe, ?Major=> Math }}.

The SolutionSpace is an intermediate state of the solution during query processing, consisting of a sequence of (preliminary) QueryResponses, each describing a unique domain. For example, the SolutionSpace of the query "What are the majors of students registered in Calculus1?" that could be represented as the sequence of two query patterns as described above could first contain two QueryResponses:

[{{?Student => Jones, ?Major=>CS},
{?Student => Smith, ?Major=>CS },
{?Student => Doe, ?Major=> Math }},
{{?Student => Jones},{?Student => Doe }}]

Each Query Response is considered to express a constraint upon the universe of possible solutions, with the actual solution being intersection of the constrained spaces. An equivalent Solution Space is therefore:

[{{?Student => Jones, ?Major=>CS},
{?Major => Math, ?Student =>Doe}}],

Part of the goal of our algorithm is to eventually reduce the Solution Space to a single Query Response like this last one.

Fig. 1 describes the top-level algorithm for answering a query. A query is answered by a process of progressively restricting the SolutionSpace by adding variable bindings (in the form of Query Responses). The initial space with no bindings ❶ represents a completely unconstrained SolutionSpace. The input query consists of a sequence of query patterns.

We repeatedly estimate the response size for the remaining query patterns ❷, and choose the most restrictive pattern ❸ to be considered next. We solve the chosen pattern by backward chaining ❹, and then merge the variable bindings obtained from backward chaining into the SolutionSpace ❺ via the restrictTo function, which performs a (possibly deferred) join as described later in this section.

When all query patterns have been processed, if the Solution Space has not been reduced to a single Query Response, we perform a final join of these variable bindings into single one variable binding that contains all the variables involved in all the query patterns ❻. The finalJoin function is described in more detail later in this section.

The estimation of response sizes in ❷ can be carried out by a combination of 1) exploiting the fact that each pattern represents that application of a predicate with known domain and range types. If these positions in the triple are occupied by variables, we can check to see if the variable is already bound in our SolutionSpace and to how many values it is bound. If it is unbound, we can estimate the size of the domain (or range) type, 2) accumulating statistics on typical response sizes for previously encountered patterns involving that predicate. The effective mixture of these sources of information is a subject for future work.

For example, suppose there are 10,000 students, 500 courses, 50 faculty members and 10 departments in the knowledgebase. For the query pattern (?S takesCourse ?C), the domain of takesCourse is Student, while the range of takesCourse is Course. An estimate of the numbers of triples matching the pattern (?S takesCourse ?C) might be 100,000

```
QueryResponse answerAQuery(query: Query)
{
   // Set up initial SolutionSpace
   SolutionSpace solutionSpace = empty; ❶

   // Repeatedly reduce SolutionSpace by
applying
   //   the most restrictive pattern
   while (unexplored patterns remain
          in the query) {
       computeEstimatesOfReponseSize
         (unexplored patterns); ❷
       QueryPattern p = unexplored pattern
           with smallest estimate; ❸

       // Restrict SolutionSpace via
       //   exploration of p

      QueryResponse answerToP =
         BackwardChain(p); ❹
      solutionSpace.restrictTo
        (answerToP); ❺
    }
   return solutionSpace.finalJoin();❻
}
```

Figure 1. Answering a Query.

TABLE I. EXAMPLE QUERY 1

Clause #	QueryPattern	Query Response
1	?S1 takesCourse ?C1	{(?S1=>s_i,?C1=>c_i)}$_{i=1..100,000}$
2	?S1 takesCourse ?C2	{(?S1=>s_j, ?C2=>c_j)}$_{j=1..100,000}$
3	?C1 taughtBy fac1	{(?C1=>c_j)}$_{j=1..3}$
4	?C2 taughtBy fac1	{(?C2=>c_j)}$_{j=1..3}$

if the average number of courses a student has taken is ten, although the number of possibilities is 500,000.

By using a greedy ordering ❸ of the patterns within a query, we hope to reduce the average size of the SolutionSpaces. For example, suppose that we were interested in listing all cases where any student took multiple courses from a specific faculty member. We can represent this query as the sequence of the patterns in Table I. These clauses are shown with their estimated result sizes indicated in the subscripts. The sizes used in this example are based on one of our LUBM benchmark [20] prototypes .

To illustrate the effect of the greedy ordering, let us assume first that the patterns are processed in the order given. A trace of the answerAQuery algorithm, showing one row for each iteration of the main loop is shown in Table II. The worst case in terms of storage size and in terms of the size of the sets being joined is at the join of clause 2, when the join of two sets of size 100,000 yields 1,000,000 tuples.

Now, consider the effect of applying the same patterns in ascending order of estimated size, shown in Table III. The worst case in terms of storage size and in terms of the size of the sets being joined is at the final addition of clause 2, when a set of size 100,000 is joined with a set of 270. Compared to Table II, the reduction in space requirements and in time required to perform the join would be about an order of magnitude.

TABLE II. TRACE OF JOIN OF CLAUSES IN THE ORDER GIVEN

Clause Being Joined	Resulting SolutionSpace
(initial)	[ ]
1	[{(?S1=>s_i, ?C1=>c_i)}$_{i=1..100,000}$]
2	[{(?S1=>s_i, ?C1=>c_i, ?C2=>c_i)}$_{i=1..1,000,000}$] (based on an average of 10 courses / student)
3	[{(?S1=>s_i, ?C1=>c_i, ?C2=>c_i)}$_{i=1..900}$] (Joining this clause discards courses taught by other faculty.)
4	[{(?S1=>s_i, ?C1=>c_i, ?C2=>c_i)}$_{i=1..60}$]

TABLE III. TRACE OF JOIN OF CLAUSES IN ASCENDING ORDER OF ESTIMATED SIZE

Clause Being Joined	Resulting SolutionSpace
(initial)	[ ]
3	[[{(?C1=>c_i)}$_{i=1..3}$]
4	[{(?C1=>c_i, ?C2=>c_i)}$_{i=1..3, j=1..3}$]
1	[{(?S1=>s_i, ?C1=>c_i, ?C2=>c'_i)}$_{i=1..270}$]
2	[{(?S1=>s_i, ?C1=>c_i, ?C2=>c_i)}$_{i=1..60}$]

```
void SolutionSpace::restrictTo
(QueryResponse newbinding)
{
   for each element oldBinding
      in solutionSpace
   {
      if (newbinding shares variables
          with oldbinding){ ❶
         bool merged = join(newBinding,
                     oldBinding, false);❷
         if (merged) {
            remove oldBinding from
              solutionSpace;
         }
      }
   }
   add newBinding to solutionSpace;
}
```

Figure 2. Restricting a SolutionSpace.

The output from the backward chaining reasoner will be a query response. These must be merged into the current SolutionSpace as a set of additional restrictions. Fig. 2 shows how this is done.

Each binding already in the SolutionSpace ❶ that shares at least one variable with the new binding ❷ is applied to the new binding, updating the new binding so that its domain is the union of the sets of variables in the old and new bindings and the specific functions represent the constrained cross-product (join) of the two. Any such old bindings so joined to the new one can then be discarded.

The join function at ❷ returns the joined QueryResponse as an update of its first parameter. The join operation is carried out as a hash join [22] with an average complexity $O(n_1+n_2+m)$ where the n_i are the number of tuples in the two input sets and m is the number of tuples in the joined output.

The third (boolean) parameter of the join call indicates whether the join is forced (true) or optional (false), and the boolean return value indicates whether an optional join was actually carried out. Our intent is to experiment in future versions with a dynamic decision to defer optional joins if a partial calculation of the join reveals that the output will far exceed the size of the inputs, in hopes that a later query clause may significantly restrict the tuples that need to participate in this join.

As noted earlier, our interpretation of the SolutionSpace is that it denotes a set of potential bindings to variables, represented as the join of an arbitrary number of QueryResponses. The actual computation of the join can be deferred, either because of a dynamic size-based criterion as just described, or because of the requirement at ❶ that joins be carried out immediately only if the input QueryResponses share at least one variable. In the absence of any such sharing, a join would always result in an output size as long as the products of its input sizes. Deferring such joins can help reduce the size of the SolutionSpace and, as a consequence, the cost of subsequent joins.

For example, suppose that we were processing a different example query to determine which mathematics courses are

TABLE IV. EXAMPLE QUERY 2

Clause	QueryPattern	Query Response
1	(?S1 takesCourse ?C1)	$\{(?S1=>s_j,?C1=>c_j)\}_{j=1..100,000}$
2	(?S1 memberOf CSDept)	$\{(?S1=>s_j)\}_{j=1..1,000}$
3	(?C1 taughtby ?F1)	$\{(?C1=>c_j, ?F1=>f_j)\}_{j=1..1,500}$
4	(?F1 worksFor MathDept)	$\{(?F1=>f_i)\}_{i=1..50}$

taken by computer science majors, represented as the sequence of the following QueryPatterns, shown with their estimated sizes in Table IV.

To illustrate the effect of deferring joins on responses that do not share variables, even with the greedy ordering discussed earlier, suppose, first, that we perform all joins immediately. Assuming the greedy ordering that we have already advocated, the trace of the answerAQuery algorithm is shown in Table V.

In the prototype from which this example is taken, the Math department teaches 150 different courses and there are 1,000 students in the CS Dept. Consequently, the merge of clause 3 (1,500 tuples) with the SolutionSpace then containing 50,000 tuples yields considerably fewer tuples than the product of the two input sizes. The worst step in this trace is the final join, between sets of size 100,000 and 150,000.

But consider that the join of clause 2 in that trace was between sets that shared no variables. If we defer such joins, then the first SolutionSpace would be retained “as is”. The resulting trace is shown in Table VI.

The subsequent addition of clause 3 results in an immediate join with only one of the responses in the solution space. The response involving ?S1 remains deferred, as it shares no variables with the remaining clauses in the SolutionSpace. The worst join performed would have been between sets of size 100,000 and 150, a considerable improvement over the non-deferred case.

TABLE V. TRACE OF JOIN OF CLAUSES IN ASCENDING ORDER OF ESTIMATED SIZE

Clause Being Joined	Resulting SolutionSpace
(initial)	[]
4	$[\{(?F1=>f_i)\}_{i=1..50}]$
2	$[\{(?F1=>f_i, ?S1=>s_i)\}_{i=1..50,000}]$
3	$[\{(?F1=>f_i, ?S1=>s_i, ?C1=>c_i)\}_{i=1..150,000}]$
1	$[\{(?F1=>f_i, ?S1=>s_i, ?C1=>c_i)\}_{i=1..1,000}]$

TABLE VI. TRACE OF JOIN OF CLAUSES WITH DEFERRED JOINS

Clause Being Joined	Resulting SolutionSpace
(initial)	[]
4	$[\{(?F1=>f_i)\}_{i=1..50}]$
2	$[\{(?F1=>f_i)\}_{i=1..50},\{(?S1=>s_j)\}_{j=1..1,000}]$
3	$[\{(?F1=>f_i, ?C1=>c_i)\}_{i=1..150} , \{(?S1=>s_j)\}_{j=1..1,000}]$
1	$[\{(?F1=>f_i, ?S1=>s_i, ?C1=>c_i)\}_{i=1..1,000}]$

```
QueryResponseSolutionSpace::finalJoin ()
{
  sort the bindings in this solution
   space into ascending order by
   number of tuples;  ❶

   QueryResponse result = first of the
     sorted bindings;
   for each remaining binding b
     in solutionSpace {
      join (result, b, true);  ❷
   }
   return result;
}
```

Figure 3. Final Join.

When all clauses of the original query have been processed (Fig. 1 ❻), we may have deferred several joins because they involved unrelated variables or because they appeared to lead to a combinatorial explosion on their first attempt. The finalJoin function shown in Fig. 3 is tasked with reducing the internal SolutionSpace to a single QueryResponse, carrying out any join operations that were deferred by the earlier restrictTo calls. In many ways, finalJoin is a recap of the answerAQuery and restrictTo functions, with two important differences:

- Although we still employ a greedy ordering ❶ to reduce the join sizes, there is no need for estimated sizes because the actual sizes of the input QueryResponses are known.
- There is no longer an option to defer joins between QueryResponses that share no variables. All joins must be performed in this final stage❷ and so the “forced” parameter to the optional join function is set to true.

IV. EVALUATION

In this section we compare our answerAQuery algorithm of Fig. 1 against an existing system, Jena, that also answers queries via a combination of an in-memory backward chaining reasoner with basic knowledge base retrievals.

The comparison was carried out using LUBM benchmarks representing a knowledge base describing a single university and a collection of 10 universities. Prior to the application of any reasoning, these benchmarks contained 100,839 and 1,272,871 triples, respectively.

We evaluated these using a set of 14 queries taken from LUBM [20]. These queries involve properties associated with the LUBM university-world ontology, with none of the custom properties/rules whose support is actually our end goal (as discussed in [23]). Answering these queries requires, in general, reasoning over rules associated with both RDFS and OWL semantics, though some queries can be answered purely on the basis of the RDFS rules.

Table VII compares our algorithm to the Jena system using a pure backward chaining reasoner. Our comparison will focus on response time, as our optimization algorithm should be neutral with respect to result accuracy, offering no more and no less accuracy than is provided by the interposed reasoner.

TABLE VII COMPARISON AGAINST JENA WITH BACKWARD CHAINING

LUBM:	**1 University, 100,839 triples**				**10 Universities, 1,272,871 triples**			
	answerAQuery		***Jena Backwd***		***answerAQuery***		***Jena Backwd***	
	response time	***result size***	***response time***	***result size***	***response time***	***result size***	***response time***	***result size***
Query1	0.20	4	0.32	4	0.43	4	0.86	4
Query2	0.50	0	130	0	2.1	28	n/a	n/a
Query3	0.026	6	0.038	6	0.031	6	1.5	6
Query4	0.52	34	0.021	34	1.1	34	0.41	34
Query5	0.098	719	0.19	678	0.042	719	1.0	678
Query6	0.43	7,790	0.49	6,463	1.9	99,566	3.2	82,507
Query7	0.29	67	45	61	2.2	67	8,100	61
Query8	0.77	7,790	0.91	6,463	3.7	7,790	52	6,463
Query9	0.36	208	n/a	n/a	2.5	2,540	n/a	n/a
Query10	0.18	4	0.54	0	1.8	4	1.4	0
Query11	0.24	224	0.011	0	0.18	224	0.032	0
Query12	0.23	15	0.0020	0	0.33	15	0.016	0
Query13	0.025	1	0.37	0	0.21	33	0.89	0
Query14	0.024	5,916	0.58	5,916	0.18	75,547	2.6	75,547

As a practical matter, however, Jena's system cannot process all of the rules in the OWL semantics rule set, and was therefore run with a simpler rule set describing only the RDFS semantics. This discrepancy accounts for the differences in result size (# of tuples) for several queries. Result sizes in the table are expressed as the number of tuples returned by the query and response times are given in seconds. An entry of n/a means that the query processing had not completed (after 1 hour).

Despite employing the larger and more complicated rule set, our algorithm generally ran faster than Jena, sometimes by multiple orders of magnitude. The exceptions to this behavior are limited to queries with very small result set sizes or queries 10-13, which rely upon OWL semantics and so could not be answered correctly by Jena. In two queries (2 and 9), Jena timed out.

Jena also has a hybrid mode that combines backward chaining with some forward-style materialization. Table VIII shows a comparison of our algorithm with a pure backward chaining reasoner against the Jena hybrid mode. Again, an n/a entry indicates that the query processing had not completed within an hour, except in one case (query 8 in the 10 Universities benchmark) in which Jena failed due to exhausted memory space.

The times here tend to be someone closer, but the Jena system has even more difficulties returning any answer at all when working with the larger benchmark. Given that the difference between this and the prior table is that, in this case, some rules have already been materialized by Jena to yield, presumably, longer lists of tuples, steps taken to avoid possible combinatorial explosion in the resulting joins would be increasingly critical.

TABLE VIII. COMPARISON AGAINST JENA WITH WITH HYBRID REASONER

LUBM	**1 University, 100,839 triples**				**10 Universities, 1,272,871 triples**			
	answerAQuery		***Jena Hybrid***		***answerAQuery***		***Jena Hybrid***	
	response time	***result size***	***response time***	***result size***	***response time***	***result size***	***response time***	***result size***
Query1	0.20	4	0.37	4	0.43	4	0.93	4
Query2	0.50	0	1,400	0	2.1	28	n/a	n/a
Query3	0.026	6	0.050	6	0.031	6	1.5	6
Query4	0.52	34	0.025	34	1.1	34	0.55	34
Query5	0.098	719	0.029	719	0.042	719	2.7	719
Query6	0.43	7,790	0.43	6,463	1.9	99,566	3.7	82,507
Query7	0.29	67	38	61	2.2	67	n/a	n/a
Query8	0.77	7,790	2.3	6,463	3.7	7,790	n/a	n/a
Query9	0.36	208	n/a	n/a	2.5	2,540	n/a	n/a
Query10	0.18	4	0.62	0	1.8	4	1.6	0
Query11	0.24	224	0.0010	0	0.18	224	0.08	0
Query12	0.23	15	0.0010	0	0.33	15	0.016	0
Query13	0.025	1	0.62	0	0.21	33	1.2	0
Query14	0.024	5,916	0.72	5,916	0.18	75,547	2.5	75,547

V. CONCLUSION /FUTURE WORK

As knowledge bases proliferate on the Web, it becomes more plausible to add reasoning services to support more general queries than simple retrievals. In this paper, we have addressed a key issue of the large amount of information in a semantic web of data about science research. Scale in itself is not really the issue. Problems arise when we wish to reason about the large amount of data and when the information changes rapidly. In this paper, we report on our efforts to use backward-chaining reasoners to accommodate the changing knowledge base. We developed a query-optimization algorithm that will work with a reasoner interposed between the knowledge base and the query interpreter. We performed experiments, comparing our implementation with traditional backward-chaining reasoners and found, on the one hand, that our implementation could handle much larger knowledge bases and, on the other hand, could work with more complete rule sets (including all of the OWL rules). When both reasoners produced the same results our implementation was never worse and in most cases significantly faster (in some cases by orders of magnitude).

In a future paper we will address the issue of being able to scale the knowledgebase to the level forward-chaining reasoners can handle. Preliminary results indicate that we can scale up to real world situations such as 6 Million triples. Optimizing the backward-chaining reasoner, together with the query-optimization reported on in this paper, will allow us to actually outperform forward-chaining reasoners in scenarios where the knowledge base is subject to frequent change

Finally, we will be working on creating a hybrid reasoner that will combine the two reasoners. We will need to be able to identify the impact on the knowledgebase specific changes have. How does the reasoner know if a fact is in the 'trusted' region or needs to be re-inferenced? How do we find facts which are revoked by a change? If we succeed we can then apply Backward reasoning only to incremental changes and periodically will do a full materialization (which does scale to billions of triples)on which we can do simple look-ups.

VI. REFERENCES

[1] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach., 1st ed. , Prentice hall, 1995, pp. 265–275.

[2] The Apache Software Foundation, Apache Jena, 2013 [retrieved: March, 2013], available from: http://jena.apache.org/.

[3] Microsoft, Microsoft Academic Search, 2013 [retrieved: March, 2013], available from: http://academic.research.microsoft.com/.

[4] Z. Nie, Y. Zhang, J. Wen and W. Ma, "Object-level ranking: bringing order to web objects", Proceedings of the 14th international World Wide Web conference, ACM Press, Chiba, Japan, May 2005, pp. 567–574, doi:10.1145/1060745.1060828.

[5] A. Doan, et al., "Community information management", IEEE Data Engineering Bulletin, Special Issue on Probabilistic Databases, vol. 29, iss. 1, March 2006, pp. 64–72.

[6] J. Tang, et al., "Arnetminer: Extraction and mining of academic social networks", Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2008), ACM Press, Las Vegas, USA, August 2008, pp. 990–998, doi: 10.1145/1401890.1401891.

[7] C. Bizer, et al., "DBpedia-A crystallization point for the Web of Data", Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, iss. 3, Sep. 2009, pp. 154–165, doi: 10.1016/j.websem.2009.07.002.

[8] F. Suchanek, G. Kasneci, G. Weikum, "Yago: A large ontology from wikipedia and wordnet", Web Semantics: Science, Services and Agents on the World Wide Web, vol. 6, iss.3, Sep. 2008, pp.203–217, doi: 10.1016/j.websem.2008.06.001

[9] B. Aleman-Meza, F. Hakimpour, I. Arpinar and A. Sheth, "SwetoDblp ontology of Computer Science publications", Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, iss. 3, Sep. 2007, pp. 151–155, doi: 10.1016/j.websem.2007.03.001.

[10] H. Glaser, I. Millard, and A. Jaffri, "Rkbexplorer. com: a knowledge driven infrastructure for linked data providers", European Semantic Web Conference, Springer-Verlag, Tenerife, Spain, Jun. 2008, pp. 797–801.

[11] A. Kiryakov, D. Ognyanov and D. Manov, "OWLIM–a pragmatic semantic repository for OWL", Proceedings of the 2005 international conference on Web Information Systems Engineering(WISE'05), Springer-Verlag , New York, USA, Nov. 2005, pp. 182-192, doi: 10.1007/11581116_19.

[12] Oracle Corporation, Oracle Database 11g R2, 2013 [retrieved: March, 2013], Available from: http://www.oracle.com/technetwork /database/enterprise-edition/overview/index.html

[13] O. Erling, I. Mikhailov, "RDF Support in the Virtuoso DBMS", Networked Knowledge-Networked Media, vol. 221, 2009, pp. 7-24, doi: 10.1007/978-3-642-02184-8_2.

[14] Y.E. Ioannidis, "Query optimization", ACM Computing Surveys (CSUR), vol. 28, iss. 1, March 1996, pp. 121-123, doi: 10.1145/234313.234367.

[15] Semanticweb.org, SPARQL endpoint, 2011 [retrieved: March, 2013], available from: http://semanticweb.org/wiki /SPARQL_endpoint.

[16] W3C, SparqlEndpoints. 2013 [retrieved: March 2013], Available from: http://www.w3.org/wiki/SparqlEndpoints.

[17] M. Stocker, A. Seaborne, A. Bernstein, C. Kiefer and D. Reynolds, "SPARQL basic graph pattern optimization using selectivity estimation", Proceedings of the 17th international conference on World Wide Web, ACM Press, Beijing, China, April 2008, pp. 595–604, doi: 10.1145/1367497.1367578.

[18] O. Hartig and R. Heese "The SPARQL query graph model for query optimization", Proceedings of the 4th European conference on the Semantic Web: Research and Applications (ESWC '07), Springer-Verlag, Innsbruck, Austria, Jun. 2007, pp. 564–578, doi: 10.1007/978-3-540-72667-8_40.

[19] W. Le, "Scalable multi-query optimization for SPARQL", 2012 IEEE 28th International Conference on Data Engineering (ICDE), IEEE Press, Washington, DC, April 2012, pp. 666–677, doi: 10.1109/ICDE.2012.37.

[20] 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, iss. 2-3, Oct. 2005, pp.158–182, doi: 10.1016/j.websem.2005.06.005.

[21] L. Ma, et al., "Towards a complete OWL ontology benchmark". Proceedings of the 3rd European conference on The Semantic Web: research and applications(ESWC'06), Springer-Verlag, June 2006, p. 125–139, doi: 10.1007/11762256_12.

[22] M. Kitsuregawa, H. Tanaka and T. Moto-Oka, "Application of hash to data base machine and its architecture". New Generation Computing, vol. 1, iss.1, March 1983, pp. 63–74.

[23] H. Shi, K. Maly, S. Zeil and M. Zubair, "Comparison of Ontology Reasoning Systems Using Custom Rules", International Conference on Web Intelligence, Mining and Semantics, ACM Press, Sogndal, Norway, May 2011, doi: 10.1145/1988688.1988708.

Enhanced Deployment Process for QoS-aware Services

Alexander Wahl and Bernhard Hollunder
Department of Computer Science
Furtwangen University of Applied Science
Robert-Gerwing-Platz 1, D-78120 Furtwangen, Germany
alexander.wahl@hs-furtwangen.de, bernhard.hollunder@hs-furtwangen.de

***Abstract*—Service-oriented architectures (SOA) are a widely used design paradigm for the creation and integration of distributed enterprise applications. The predominant technology used for implementation are Web services. In the business domain, Web Services must be equipped with quality of service (QoS) attributes, like, e.g., security, performance, etc. WS-Policy standard provides a generic framework to formally describe QoS. Verification and enforcement of such formally described QoS require corresponding QoS modules, e.g., handlers, to be installed at the runtime environment. In this work, we provide an approach that enhances the current deployment process for Web services. The aim is to provide a sophisticated process that not only deploys Web Services and formal QoS descriptions, but to guarantee that the desired QoS are verified and enforced in the runtime environment. We therefore introduce additional steps for the analysis of WS-Policy descriptions, the identification of corresponding handlers, and their installation at the runtime environment. As a result, a comprehensive, automated deployment process is created, and the fulfillment of an overall WS-Policy description is ensured.**

***Keywords*-Service-oriented architecture; QoS-aware service; Deployment.**

I. Introduction

The usage of distributed systems is nowadays widespread. With the increased availability of networks, especially the internet, distributed systems find application in business domains as well as in private environments. Typically, some desired distant functionality is used following the client-server model. The Service-oriented Architecture (SOA) paradigma is a well-known paradigma to form such a system: business functionality is implemented as a service and offered to service consumers. The communication takes place over a, possibly public, network.

Beside the offering of pure functionality, the fulfillment of non-functional aspects that relate to Quality of Service (QoS), is an important factor. For example, for e-payment security and transactional behavior aspects are obviously crucial. Beside these exemplary QoS several others exist, that apply to the domain of distributed systems, respectively SOA, such as response time, availability, cost, usage control, roles and rights, etc.

In a technical environment, like a SOA infrastructure, it is worthwhile that desired QoS are analyzed and enforced in an automated manner by the infrastructure itself. In order to achieve such an automatism these QoS first have to be formally described. Next, the formal description is related to the desired services, following the separation of concerns (SoC) principle [1]. In other words, the functionality implementation is separated from the QoS implementation. The functionality is implemented using some programming language, like Java, C++ or C#. For the QoS implementation usually declarative languages, so-called policy languages, are used. The resulting service is then called a QoS-aware service.

A well-known and widely used language to formally describe QoS is WS-Policy [2]. In the context of WS-Policy a policy is a collection of policy alternatives. Each policy alternative is a set of policy assertions. The policy assertions are used to express QoS. It should be noted, that WS-Policy does not provide concrete assertions, but is a more general framework. Concrete assertions are introduced in related specification, like, e.g., WS-SecurityPolicy [3]. With WS-Policy, other domain-specific assertions can be defined.

Creating appropriate formal descriptions of a QoS is in the repsonsibility of the service developer. This is no easy task, and a deep knowledge on several different models, languages and technologies is a must. This is especially true with QoS for whom no standardized specifications, frameworks and approaches are available.

The formal description of a QoS is just half the way. The validation and enforcement of such a QoS description is in the responsibility of the service runtime environment. However, it has to be ensured, that the specified QoS of a service is understood and enforced by that runtime environment. One can think of different approaches to achieve that. In this work, we will enforce QoS by using specialized service agents, so-called handlers. These handlers are installed, configured and invoked at the runtime environment. It has to be mentioned, that this approach does not apply to all QoS, but to a significant set of of QoS. Again, a comprehensive knowledge on the infrastructure used, the service provider and different used technologies, e.g. frameworks, is necessary.

Figure 1 gives an overview on a typical process currently used to create a QoS-aware service. The process used so far attaches a QoS description (using, e.g., WS-Policy) to a service implementation. The resulting artifact is then

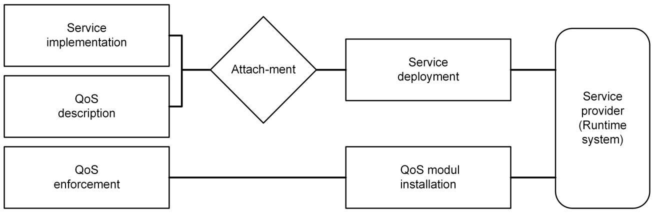


Figure 1. QoS-aware service development, deployment and installation process.

transferred to the service provider. Todays process does not check the availability of corresponding QoS modules at the runtime environment, i.e., the service provider. It does not include an analysis of the QoS description to identify, install and configure the QoS modules, i.e., handlers, that are necessary to enforce the QoS. The solution of this deficit is enhance the current deployment process by an automated module installation, which is based on the analysis of a QoS description. By that means it can be guaranteed, that a QoS description are verified and enforced during runtime.

In this work, we integrate additional steps to the deployment process for policy analysis, and QoS module identification, installation and configuration. The aim is to provide a comprehensive, streamlined and significantly eased deployment process for QoS-aware services. As benefit

1) the development process of QoS-aware WS is improved by additional steps that identify and install necessary QoS modules, and
2) the availability and usage of uniform (and thereby more interoperable) QoS descriptions and corresponding handlers is increased.

This paper is organized as following: In Section II the problem is described in more detail, and a solution strategy is introduced. Section III provides a general overview on our approach, which is explained more precisely in Section IV. Section V presents a prototypic implementation of our solution. A discussion on related work is given in Section VI, followed by a conclusion in Section VII.

II. Problem Description and Solution Strategy

The development of QoS-aware services is a demanding task. Beside the implementation of the service, the desired QoS need to be formally described. Both, service and QoS description need to be bound, and afterwards be deployed to the runtime system. Except for a few QoS, that are supported by current frameworks, enforcement modules for the desired QoS also need to be implemented, and afterwards installed at the runtime system. Overall, a developer needs to have extensive knowledge of available languages and applicable technologies in order to realize a QoS-aware service.

A more sophisticated approach is desirable. Such an approach first enables to describe QoS, including non-standardized ones, at a higher level of abstraction, and to generate appropriate formal representation in an automated manner. Next, it applies the generated QoS representation to the services under development, and identifies appropriate QoS modules. Finally, it gathers and installs – if necessary – the required QoS modules at the infrastructure.

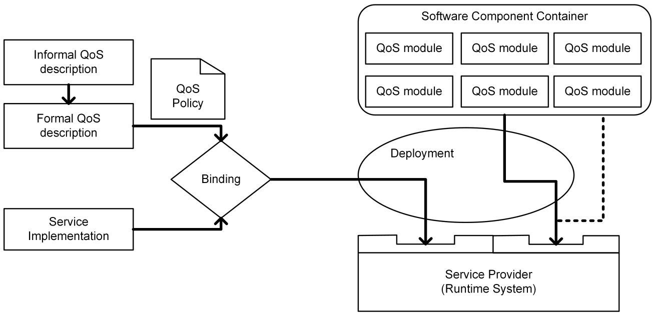


Figure 2. Enhanced service deployment overview.

This approach can be divided into two parts. The former part is already covered by Al-Moayed, Hollunder and Wahl [5], who provide a solution to specify QoS at an higher level of abstraction and to generate corresponding WS-Policies descriptions. In this work, we enhance this approach by the latter steps described before: The generated QoS description is analyzed, and corresponding QoS modules are identified. Afterwards these QoS modules are retrieved from a dedicated container component (i.e. the Software Component Container), configured, and finally installed at the service provider. Figure 2 visualizes this approach.

III. Approach

In this section, we describe the deployment process steps in more detail: the analysis of a formal QoS description with regard to QoS module identification, the QoS module identification itself, and the QoS module installation (see Figure 1).

The introduced approach is part of a more general approach described in Hollunder, Al-Moayed and Wahl [6]: A Tool Chain for Constructing QoS-aware Web Services.

The formal description of a desired QoS, i.e., QoS description, is realized with WS-Policy. In Al-Moayed, Hollunder and Wahl [5] a QoS is specified using a model-based approach. On that QoS model model-to-model and model-to-code transformations are performed that finally create a formal QoS description based on WS-Policy.

In a first step, the generated WS-Policy description is processed. The aim is to identify all occuring assertions. Assertions are the basic building blocks of a WS-Policy. These assertions reflect QoS. For each assertion a QoS module must be available, and the QoS module verifies and enforces the QoS. The identification of such a specific module requires knowledge on which assertions a QoS module is capable of. In our approach, we introduce the Software Component Management Unit, whose responsibility is to i) store and manage all available QoS modules, and ii) to store the information how assertions are implemented by each QoS module.

For each identified assertion of the WS-Policy description, a corresponding QoS module is searched within the Software Component Management Module. Once the assertions are related to a corresponding QoS module, the required QoS modules to enforce the given WS-Policy are known. With that information, the runtime environment can be equipped with these QoS modules. This will ensure, that the complete WS-Policy description is verified and enforced at the runtime environment.

IV. Enhanced Deployment Process

In this section we will describe the individual steps and components of the approach presented in the previous section.

A. Analysis of a formal QoS description

Consider a formal QoS description, e.g., based on WS-Policy. Listing 1 shows an example, which was initially described in [5]. Line 1 defines a policy description with Id `CalculatorConstraintPolicy`. Line 2 specifies, that the following are policy alternatives, which is equivalent to OR. Line 3 introduces a set of policy assertion, which equals AND. Lines 4 and 5 are assertions that specify a QoS constraint – a range of numbers – with two QoS parameter `wscal:minInt` and `wscal:maxInt`. The remaining lines are closing tags for lines 1-3.

Upon closer examination, `wscal:minInt` and `wscal:maxInt` in lines 3 and 4 concrete assertions. Both have to be interpreted by an appropriate handler (e.g., http handler or SOAP handler). Since these assertions do not belong to any known WS-Policy related specification, the handler have to be developed from scratch. Same is true for any other custom assertion.

```
1 <wsp:Policy wsu:Id=”CalculatorConstraintPolicy”>
2  <wsp:ExactlyOne>
3   <wsp:All>
4    <wscal:minInt Number=”−32768”> </wscal:minInt>
5    <wscal:maxInt Number=”32767”> </wscal:maxInt>
6   </wsp:All>
7  </wsp:ExactlyOne>
8 </wsp:Policy>
```

Listing 1. Calculator service WS-Policy description.

As described before, the initial step is to identify all assertions within the WS Policy. In our example, these are the two assertions `wscal:minInt` and `wscal:maxInt` in line 4 and 5. Therefore, in our example the result of the analysis step is a list of two elements.

B. Software Component Container

The Software Component Container is a central component of the approach. Its main purpose is to contain all available QoS modules. For each such QoS module there is a relation to the WS-Policy assertions it implements. This relation is also stored in the container.

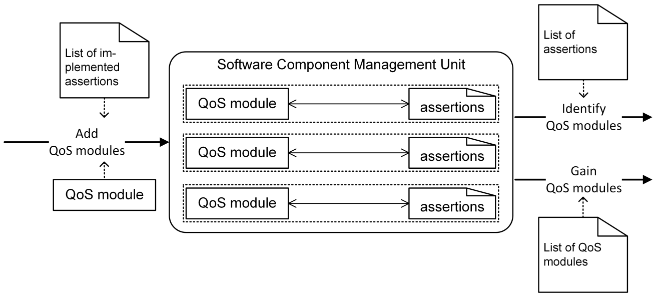


Figure 3. Software Component Container.

There are three interfaces at the Software Component Container (see Figure 3). The first interface provides a means to add QoS modules to the container. This interface requires information on the implemented assertions and the QoS module. Via a second interface a list of corresponding QoS modules for a given list of assertions can be retrieved. Finally, QoS modules can be gained from the Software Component by a third interface.

C. QoS module identification

QoS modules are well-defined software components that enforce a desired QoS. In this work, we focus on handlers, which are one category of QoS modules. The term QoS module therefore is equivalent handler. However, this approach is not limited to the handler approach.

This step identifies the QoS modules needed to verify and enforce the formalized QoS described in the WS-Policy file. Input for this step is the list of QoS assertions described before. For each entry of that list the Software Component Container is inquired. Remind that the Software Component Container is aware of all assertions implemented by any of its registered QoS modules. If an assertion is found at the Software Component Container, a corresponding QoS module is available, and the module is stored in a list of necessary QoS modules. Otherwise, no appropriate QoS module is available, and the assertion cannot be verified and enforced. In that case the assertion is added to a list of unresolved assertions. With unresolved assertions there are different options, ranging from canceling the enhanced deployment process up to inform the developer at the end of the deployment process.

In summary, the QoS module identification step is able to identify QoS modules, that are needed to enforce an QoS, based on the assertions used within the WS-Policy. It further enables to identify assertions, where do not exist corresponding QoS modules. With that, the developer of a QoS-aware service is able to recognize in a proactive manner, which assertions can or cannot be handled by the runtime environment, reflected by i) a list of QoS modules to be installed, and ii) a list of unresolved assertions.

D. QoS module installation

Once the QoS modules are identified, the WS runtime environment is checked, if these modules are already installed. If there are modules missing, they are collected from the Software Component Management Unit, configured and installed. Using WS technology, the runtime environment is typically an application server. Such an application server usually provides a management API, that can be accessed to install additional modules, like the QoS modules. Once all QoS modules are installed, it is assured that the overall WS-Policy description given can be fulfilled.

E. Automation

Up to today the deployment process of a QoS-aware service mainly includes steps for buidling, packaging and installing in some container, as described elsewhere, e.g., for WSIT [7]. As visualized in Figure 1, identification, configuration and installation of necessary QoS modules is not included in the deployment process. In the preceding paragraphs we identified the steps that are to be performed in order to be sure that a given WS-Policy can be completely handled. We state that each of these steps can be automated. There is no need for user interaction at each of these steps – even with unresolved assertions. We therefore argue to enhance the current deployment process with the described steps.

V. Proof of Concept

For a proof of concept, we focused on SOA using Java-based infrastructure and technologies. In detail:

- NetBeans IDE [8]
- Java API for XML Web Services (JAX-WS) [9]
- Glassfish application server [10]
- Eclipse IDE [11]
- Apache Ant [12]
- Apache Neethi [13]
- Apache Subversion [14]
- MySQL Community Edition [15]

NetBeans IDE is used to create Web Services based on JAX-WS technology. We further use the Glassfish application server. It can be registered to the NetBeans IDE as deployment destination. NetBeans IDE uses Apache Ant to implement the deployment process, as described in the WSIT Tutorial [7]. The corresponding Apache Ant build files, for building the Web Archive (WAR) and for deployment, are generated by NetBeans.

Eclipse IDE is used to create a formal QoS description based on WS-Policy. We use Eclipse IDE due to the fact, that the prototypic tool of Al-Moayeds approach is a Eclipse plugin.

The description of the WS interface is realized using the Web Service Description Language (WSDL). WSDL may refer to a WS-Policy description. When a WS is invoked, the WS runtime environment recognizes the existence of a policy and delegates the request to the installed handlers. A handler then processes the request according to the policy assertions it is responsible for.

```
<target description="Build Web Archive (WAR)."
        name="dist">
  <jar jarfile="dist/CalculatorService.war">
      <fileset dir="web" />
  </jar>
</target>
```

Listing 2. Building a Web Archive (WAR) using Apache Ant.

Apache Ant is used to automate the creation of the WAR and the deployment. Listing 2 displays the corresponding target `dist`, which uses the task `jar` to create a WAR named `CalculatorService.war` in folder `dist`.

```
<target description="Deploy Web Archive (WAR)."
        name="deploy">
  <get src="http://localhost:4848/__asadmin/deploy?
            path=dist/CalculatorService.war"/>
</target>
```

Listing 3. Deployment of a Web Archive (WAR) using Apache Ant.

Listing 3 shows the Ant target `deploy` to deploy the WAR file to a Glassfish application servers. It uses the Administration Console running on `localhost:4848`, and invokes the `asadmin` command with parameter `deploy`. Afterwards the `CalculatorService.war` is available.

```
<target name="identify-handler">
  <java jar="qos-module-identification.jar">
    <arg value="in=policy.xml"/>
    <arg value="out=handler.xml"/>
    <arg value="out=unresolved-assertions.xml"/>
  </java>
</target>

<target name="install-handler">
  <java jar="qos-module-installation.jar"/>
    <arg value="in=handler.xml"/>
  </java>
</target>
```

Listing 4. Steps introduced in Enhanced Deployment Process.

Between these two steps, `dist` and `deploy`, we introduce further steps that enhance the deployment process by the step for WS-Policy analysis, handler identification and installation, as described in Section IV. In Listing 4 the two introduced targets are shown. The first target, `identify-handler`, invokes `qos-module-identification.jar`, which implements the WS-Policy analysis and handler identification steps; `qos-module-installation.jar` implements the handler installation, which is invoked by `install-handler`.

Within `qos-module-identification.jar` Apache Neethi, an open-source implementation of WS-Policy, is used to parse a WS-Policy description and to identify its assertion. Afterwards, the Software Component Container (described later) is invoked for each assertion to identify the implementing QoS module. To compare the assertions the policy intersection algorithm of WS-Policy is used. If a match of assertions is found, the corresponding QoS module is stored. Otherwise the assertion is added to the list of unresolved assertions.

The Software Component Container responsibility is to administer the individual QoS modules, and to track the assertions implemented by a QoS module. We use a software versioning and revision control system, Apache Subversion, to version each QoS module. For each QoS module metadata is stored. Beside others, this metadata mainly consists information on the assertions implemented with a QoS module, author, version, etc. These data are saved using a MySQL database. Both, Apache Subversion and MySQL come with APIs for Java, which enables to implement a dedicated component for QoS module identification.

A further Java-based component, implemented within `qos-module-installation.jar`, is used to gather and install the identified handler within the Glassfish application server. This component checks out the handler from the repository using the Subversion API. Afterwards, the QOS modules are installed. We use the Applicationserver Management eXtensions (AMX) and Java Management Extensions (JMX) to perform this step.

The proof of concept showed that an automated identification of handler based on assertions in a WS-Policy description, and an installation of these handler is feasible. The approach ensures that an overall WS-Policy description given can be fulfilled. But it showed that the execution of the Enhanced Deployment Process requires administrative access to the application server.

VI. Related Work

There are books and papers that describe QoS-aware WS, the deployment process, and the use of QoS components, i.e., handlers. However, the identification and installation of QoS modules is so far a manual step.

In Erl's book [4] service deployment is a phase of the SOA delivery lifecycle. In this stage distributed components, service components, service interfaces, and any associated middleware products are installed and configured on the production server. In another book [16], he describes how to add WS-Policy descriptions to a WSDL.

WS-Policy is widely used to formalize QoS. Hollunder [17] discusses the introduction of an operator in WS-Policy for conditional assertions. He further describes the implementation of a corresponding policy handler based on the Apache Axis framework. Mezni, Chainbi and Ghedira [18] extend WS-Policy to specify services related data in order to enable for policy-based self-management and to describe autonomic Web services. Mathes, Heinzl and Freisleben [19] extend WS-Policy to introduce time-dependant policy descriptions, which allows to specify time constraints on the validity of a policy description.

In this work assertions are matched using the policy intersection algorithm. Hollunder [20] presents a new approach to determine the compatibility of policies that operates not only syntactically but also takes into account the semantics of assertions and policies. Brahim, Chaari, Jemaa and Jmaiel [21] present a sematic approach to match WS-SecurityPolicy assertions.

Al-Moayed, Hollunder and Wahl [5] introduce a model-based approach to create a policy description based on WS-Policy. They use a meta-model introduced by Malfatti [22].

A description of the deployment process using Apache Ant is provided in the WSIT Tutorial [7]. Another framework, Apache CXF [23], also uses Apache Ant to implement the deployment process.

VII. Conclusion

Designing and implementing QoS for a SOA is a demanding task. Up to now just a few QoS are supported by IDEs, tools and frameworks. But there are several QoS that are still implemented in a proprietary manner. Also, for the verification and enforcement corresponding QoS modules need to be developed. As a result numerous different implementations for each QoS exists, which are usually not interoperable.

Automated deployment processes for Web Service that use WS-Policy to describe QoS are established and widely-used. However, identification, configuration and installation of QoS modules, e.g., handlers, at the runtime environment is still performed manually.

In this work, we proposed a way to identify the QoS modules needed to enforce a WS-Policy description. We further introduced a component i) to manage available QoS modules including different versions, ii) to store the implemented constraints of each QoS module, and to iii) identify the QoS modules necessary to verify and enforce an overall WS-Policy. We further introduced a component that is able to gather the desired QoS modules and to install them at the runtime environment.

Finally, we showed that the deployment process can be enhanced to identify and install missing QoS modules automatically, to identify assertions that cannot be implemented at all due to unavailable QoS modules.

Further, by using a central, dedicated Software Component Management we expect to improve the availability of QoS modules and corresponding QoS descriptions over time by publicly offering such a unit.

The benefit of this work is the consolidation of two separate processes that are both necessary to successfully implement QoS-aware Web Services. One is the deployment process of Web Services and WS-Policy descriptions. The other is the installation of handler at the infrastructure, which also has been automated. The introduction of a dedicated unit that manages QoS modules, relates them to the implemented assertions is a valuable progress.

But there are still challenges. We plan to investigate further QoS with regard to implementation strategies, and on new technologies to support this implementation. Further, we want to improve the availability of QoS. Finally the comprehensive tool chain for QoS-aware Web Services is further improved.

ACKNOWLEDGMENT

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] S. Robertson and J. Robertson, *Mastering the requirements process*, 3rd ed., P. Education, Ed. Addison-Wesley, 2012.

[2] A. Vedamuthu, D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, and U. Yalcinalp, “Web services policy 1.5 - framework,” World Wide Web Consortium, Tech. Rep., 2007, last access: 15. Jan. 2013. [Online]. Available: http://www.w3.org/TR/ws-policy/

[3] K. Lawrence and C. Kaler, “WS-SecurityPolicy 1.2,” OASIS, Tech. Rep., Feb. 2009, last access: 15. Jan. 2013. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/ws-securitypolicy.html

[4] T. Erl, *Service-oriented architecture*, 9th ed. Upper Saddle River, NJ [u.a.]: Prentice-Hall, 2005.

[5] A. Al-Moayed, B. Hollunder, A. Wahl, and V.Sud, “Quality attributes for web services: A model-based approach for policy creation,” *International Journal on Advances in Software*, vol. 5, no. 3&4, pp. 166–178, Dec. 2012. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=soft_v5_n34_2012_2

[6] 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 Research Directions*. IGI Global, 2012, pp. 189–211. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-794-4.ch009

[7] Oracle, “The WSIT tutorial,” last access: 15. Jan. 2013. [Online]. Available: http://docs.oracle.com/cd/E17802_01/webservices/webservices/reference/tutorials/wsit/doc/index.html

[8] ——, “Netbeans IDE.” [Online]. Available: http://netbeans.org/

[9] java.net, “JAX-WS,” last access: 15. Jan. 2013. [Online]. Available: http://jax-ws.java.net/

[10] ——, “Glassfish,” last access: 15. Jan. 2013. [Online]. Available: http://glassfish.java.net/de/

[11] The Eclipse Foundation, “Eclipse IDE,” last access: 15. Jan. 2013. [Online]. Available: http://www.eclipse.org/downloads/moreinfo/jee.php

[12] The Apache Software Foundation, “The apache ant project,” last access: 15. Jan. 2013. [Online]. Available: http://ant.apache.org/

[13] ——, “The apache neethi project,” last access: 15. Jan. 2013. [Online]. Available: http://ws.apache.org/neethi/

[14] ——, “The apache subversion project,” last access: 15. Jan. 2013. [Online]. Available: http://subversion.apache.org/

[15] Oracle, “Mysql community edition.” [Online]. Available: http://www.mysql.com/

[16] T. Erl, Ed., *Web service contract design and versioning for SOA*, ser. The @Prentice Hall service-oriented computing series from Thomas Erl. Upper Saddle River, NJ [u.a.]: Prentice Hall, 2009.

[17] B. Hollunder, “Ws-policy: On conditional and custom assertions,” in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, july 2009, pp. 936 –943.

[18] H. Mezni, W. Chainbi, and K. Ghedira, “Aws-policy: An extension for autonomic web service description,” *Procedia Computer Science*, vol. 10, no. 0, pp. 915 – 920, 2012, ¡ce:title¿ANT 2012 and MobiWIS 2012¡/ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050912004796

[19] M. Mathes, S. Heinzl, and B. Freisleben, “Ws-temporalpolicy: A ws-policy extension for describing service properties with time constraints,” in *Computer Software and Applications, 2008. COMPSAC ’08. 32nd Annual IEEE International*, 28 2008-aug. 1 2008, pp. 1180 –1186.

[20] B. Hollunder, “Domain-specific processing of policies or: Ws-policy intersection revisited,” in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, July, pp. 246–253.

[21] M. B. Brahim, T. Chaari, M. B. Jemaa, and M. Jmaiel, “Semantic matching of ws-securitypolicy assertions,” in *Service-Oriented Computing - ICSOC 2011 Workshops*. Springer Berlin Heidelberg, 2012, pp. 114–130. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31875-7_13

[22] D. Malfatti, “A Meta-Model for QoS-Aware Service Compositions,” Master’s thesis, University of Trento, Italy, 2007.

[23] The Apache Software Foundation, “Apache CXF,” last access: 15. Jan. 2013. [Online]. Available: http://cxf.apache.org/

A Trust-based Model for Quality of Web Service

Bo Ye, Anjum Pervez, Mohammad Ghavami
the Faculty of Engineering, Science and Built Environment
London South Bank University
London, UK
{yeb,perveza,ghavamim}@lsbu.ac.uk

Maziar Nekovee
BT Research,
Polaris 134 Adastral Park,
Martlesham, Suffolk, UK
maziar.nekovee@bt.com

Abstract—**In order to choose the best services among various services across the world, how much a service can be trusted is increasingly important for service consumers. In addition, if services are invoked by machines, it is increasingly crucial that the trust in services can be calculated automatically. However, most existing approaches are based on the assumption that the trust value can be provided by consumers, but 'how' is not solved. In this paper, criteria of Quality of Service (QoS) are classified into different groups, and an automatic trust calculation is introduced. After that, an approach based on the Kalman Filter is presented to filter out malicious values and to estimate real values. Through aggregating the values provided by other consumers, the value of the trust in different QoS criteria can be obtained. Finally, experiments are carried out to access the validation and robustness of our model.**

Keywords-Web service; Quality of Service; Trust; Kalman Filter

I. Introduction

While more and more systems are developed based on Service and Cloud Computing, more and more different kinds of services are going to emerge on the Internet. Because of various users' requirements and many service attributes, selection of a proper service is not easy for a user in Service Oriented Architecture (SOA) systems. Currently, most approaches are based on Quality of Service (QoS) to select services.

Among these QoS criteria, reputation is a really important one because service consumers can access a large number of services providing identical or similar functions. Most providers publish values of their services' QoS. However, how much these values can be trusted is really crucial. Electronic commerce has a similar issue, and various electronic markets and online electronic commerce companies have built reputation systems, e.g., Amazon, eBay, Yahoo!, Slashdot. Many researchers have considered trust-based system, an effective way to identify malicious consumers to minimize their threat and protect the system from possible misuses and abuses.

Therefore, it is crucial to measure web service trust and many researchers have proposed various approaches using different techniques. Although many efficient and robust measure solutions [1]–[6] has been proposed in previous research, these approaches still mostly have following weaknesses.

First of all, most approaches measure service reputation based on the feedback provided by human, and therefore it is difficult to ensure the accuracy. This kind of systems cannot be built without humans participating. Due to different abilities and knowledge of human, it is impossible for them to provide the same feedback, even if they use the identical service. Hence, it is necessary to build an automatic trust measure system.

Secondly, existing trust measure solutions mostly collect feedbacks only between service consumers and providers, but rarely use information among service consumers. Most of such systems are centralized, and not all systems have a center server. Hence, how a new service consumer can obtain the trust level of a service in distributed systems needs to be considered.

In addition, due to existing malicious service consumers, how those malicious ones can be filtered out becomes increasingly crucial. Malicious consumers might provide malicious values to falsely improve the trust, or to degrade the trust in certain service providers for commercial benefits.

To address the weaknesses above, an approach to measure the trust both in service providers and consumers has been proposed. This approach first groups QoS criteria, and then measures the trust in each QoS criterion of a service based on the characteristics of QoS criteria. The measure of trust in services has also been divided into two main stages, including Time Domain and Aggregation Domain, because a service consumer can measure the trust in service providers at different domains. First, it may invoke a service many times, so that it can measure the trust based on its own observation at Time Domain. All data obtained itself at Time Domain, and therefore it is unnecessary to filter out any information. Second, it may also obtain the trust by using the data from others, called as Aggregation Domain. At this stage, the consumer not only needs to aggregate all information, but also has to filter out malicious data.

Compared to the existing approaches, our main contributions have been summarised as follows:

1) Quality of Web Service Criteria have been grouped into several classes based on their characteristics, and the stages of trust measure have been classified into two main domains. A trust measure approach has been proposed to overcome the weaknesses mentioned above.
2) Both negative and positive malicious values can be detected by our approach, and the trust in both service providers and consumers has been measured, which makes the measure more reliable. Trust in consumers reflects its trust level from another one's perspective.
3) At Aggregation Domain, when a service consumer ag-

ISBN: 978-1-61208-270-7

gregates the information from numerous other service consumers, it uses the trust in other consumer X to weight the trust in a service provider from X.

The rest of this paper is organized as follows. The major related research has been introduced in Section I. Section III presents how quality criteria are grouped and how the values of trust in quality criteria and reference trust are calculated automatically. Section IV describes how malicious values are detected and how different values from a variety of consumers are aggregated to calculate the value of trust. The model is evaluated by carrying out different experiments in Section V. Finally, the conclusion is given in the Section VI.

II. Related Work

The characterization of reputation systems and threats facing them from a computer science perspective can be found in [4]. Wang and Lin [5] reviewed the reputation-based trust evaluation mechanisms in literature and outline some trust issues in e-commerce environments. An overview of existing and proposed systems that can be used to derive measures of trust and reputation for Internet transactions was given in [6].

Ardagna and Pernici [7] proposed a modelling approach to service selection problem, but trust is just one of those considered quality criteria. Although this approach is effective to select a service for a consumer, it did not focus on the detection of malicious consumers. Then if malicious consumers exist, it may not select appropriate services for consumers.

In [8], the reputation was modelled as a three-dimension belief (b, d, u), which represent the positive, negative and uncertain probabilities. In [9]–[11], a Bayesian reputation approach was proposed to calculate the reputation value based on the beta probability density functions (PDF). In [10], intuitive parameters needs to be tuned manually without guarantee of any quantitative confidence. Wang, Liu and Su [12] proposed a general trust model for a more robust reputation evaluation. The trust in [8]–[10] was modelled as predicted probability values, but prediction variance was ignored by them, which was considered in [12]. However, all these models only use feedbacks between service providers and consumers, but did not use feedbacks among service consumers.

Based on reputation Corner et al. [13] proposed a trust management framework that supports the synthesis of trust-related feedback from multiple services while also allowing each entity to apply various scoring functions over the same feedback data for personalized trust evaluations. However, this approach is based on the assumption that consumers do not mask their malicious behaviour, meaning that it is hard to detect those malicious service consumers that behave well until they gain good trust values and then behave maliciously. Xiong et al. and Srivatsa et al. [14], [15] measured the trust by the use of personalized similarity between itself and other peer X to weight feedback from X.

However, all these models were built based on the assumption that a consumer can provide a trust value in the service. This is the first weakness summarized in Section I. How a trust value can be automatically calculated is ignored in these models.

III. Quality Criterion

In this section, Quality Criterion and relevant concepts are introduced first, and then, how trust values can be calculated automatically is presented.

Definition 1. ***Quality Criterion:*** *This encompasses a number of Quality of Service (QoS) properties used to evaluate a web service. The following is a brief explanation of criteria:*

1) **Price (P):** The price of a web service is a sum of the invoked operations' prices of the web service.
2) **Response Time (RT):** The estimated time between when a request is sent and when a result is received.
3) **Availability (A):** The probability that a web service is available.
4) **Success Rate (SR):** The rate of a web service's ability to response to requests successfully.

Based on the way how it affects the overall QoS of a service, Quality Criterion can be classified as either **Positive Criterion**, whose increase benefits the overall QoS, or **Negative Criterion** whose decrease benefits the overall QoS. Based on the nature of a criterion, criteria fall into two major classes:

1) **Ratio Criterion:** The value of a criterion can be expressed as a ratio, and their values can also be directly used as the trust in the criterion, e.g., availability, success rate, etc. Please note that Ratio Criteria are not the criteria whose values obtained from providers are rate. For example, compensation rate's values are rate. However, it is not a ratio criterion.
2) **Non-ratio Criterion:** The value of a criterion can not be expressed by a ratio, e.g., price, response time, etc.

Because there are different ways to obtain the value of a criterion, there are two major classes of criterion value:

1) **Published Value of a Criterion:** The value of a criterion is published by service provider. This can be updated by providers at any time.
2) **Actual Value of a Criterion:** The value of a criterion is obtained by service consumers after invoking a service, which may be different from Published Value. For instance, a provider may publish $40ms$ as a service's response time, but the actual response time may be $43ms$ when the service is invoked.

Definition 2. ***Trust:*** *To simplify description, in this paper, 'Trust' is used as a property of a service, denoted by T.*

For example, a consumer A has a trust in another consumer B, meaning that A knows how much he can trust B. Trust can be classified as either criterion or reference trust, on the basis of trust purpose.

1) **Criterion Trust:** A consumer A's trust in a criterion C of a service S provided by a service provider P, denoted by $T(A \rightarrow P.S.C)$. It identifies how much a Published Value of $P.S.C$ can be trusted.

2) **Reference Trust:** Consumer A's trust in consumer B's capacity of referring to other consumers' ability to do something, defined by $T(A \rightarrow B)$. Please note that a service provider can also have a reference trust, because the service provider can also be a service consumer, recommending another service provider.

Based on the ways how it affects the overall trust, a trust can be divided into **Positive Part**, which increases the trust, and **Negative Part**, which decreases the trust.

Similarly, the actual value of a criterion can also be classified as either **Positive Actual Value**, which increases the trust of the criterion, or **Negative Actual Value**, which decreases the criterion's trust.

A. Criterion Trust Calculation

$T(A \rightarrow S.C)_j$ represents A's trust in service S's criterion C after j^{th} time user A invokes web service S. c represents the published value of $S.C$, while c_j is the actual value obtained by the user after j^{th} time invoking service S.

Suppose Criterion C is a negative one, meaning that the decrease of this criterion benefits the trust, then $T(A \rightarrow S.C)_j$ is calculated by the following equations.

The number of positive C values, num_j^{po}, is calculated by:

$$\mathrm{num}_j^{po} = \begin{cases} \mathrm{num}_{j-1}^{po} + 1 & c_j \leq c \\ \mathrm{num}_{j-1}^{po} & c_j > c \end{cases} \tag{1}$$

The number of negative C values, num_j^{ne}, is computed by:

$$\mathrm{num}_j^{ne} = \begin{cases} \mathrm{num}_{j-1}^{ne} & c_j \leq c \\ \mathrm{num}_{j-1}^{ne} + 1 & c_j > c \end{cases} \tag{2}$$

The following equation is used to calculate the value of positive part of C's trust,

$$T_j^{po} = \begin{cases} \sqrt{\frac{\mathrm{num}_{j-1}^{po}(T_{j-1}^{po})^2 + (1-\frac{c_j}{c})^2}{\mathrm{num}_j^{po}}} & c_j \leq c \\ T_{j-1}^{po} & c_j > c \end{cases} \tag{3}$$

Value of negative part of C's trust,

$$T_j^{ne} = \begin{cases} T_{j-1}^{ne} & c_j \leq c \\ \sqrt{\frac{\mathrm{num}_{j-1}^{ne}(T_{j-1}^{ne})^2 + (1-\frac{c_j}{c})^2}{\mathrm{num}_j^{ne}}} & c_j > c \end{cases} \tag{4}$$

At last, $T(A \rightarrow S.C)_j$ is calculated by

$$T(A \rightarrow S.C)_j = 1 + T_{j-1}^{po} - \frac{\mathrm{num}_j^{ne}}{\mathrm{num}_j^{ne} + \mathrm{num}_j^{po}} \cdot T_j^{ne} \tag{5}$$

Please note that the equal values to the Published Value are always classified as positive values.

B. Reference Trust Calculation

A service consumer A's reference trust in another consumer B is used to identify how much B can be trusted by A. Using the value of trust in B, A can know how much he can trust the services or other consumers referred by B.

Because a service provider can provide various services and an identical service can be provided by a number of service providers, $P.S.C$ is used to denote a service S's criterion C provided by a service provider P. $T(A \rightarrow P.S.C)$ represents the value of A's trust value in $P.S.C$, while similarly $T(B \rightarrow P.S.C)$ represents the value of B's trust value.

The set of trust's values obtained by service users can be viewed as a multidimensional space and each user can be a point in the space. Hence, A's trust $T(A \rightarrow B)$ in service consumer B can be calculated by the geometric distance between the points as follows

$$T = 1 - \sqrt{\frac{\sum_P \sum_S \sum_C (T(A \rightarrow P.S.C) - T(B \rightarrow P.S.C))^2}{|T(A \rightarrow P.S.C)|}} \tag{6}$$

Their values are more similar, meaning that their experience is more similar, and then, A can trust B more.

C. Trust Transitivity

If a service consumer A needs to know how much he can trust in criterion C of service provider S, but he has no information about it. However, B has trust in criterion C of service provider S, and A knows how much he can trust B. Then A's trust in $S.C$ can be defined by:

$$A \rightarrow S.C = (A \rightarrow B) \cap (B \rightarrow S.C) \tag{7}$$

In this equation, the symbol $\cap$ does not mean that $A \rightarrow B$ and $B \rightarrow S.C$ intersect. It means that based on the trust's transitive property, A's trust in $S.C$ can be derived by A's reference trust in B and B's criterion trust in $S.C$.

It is common to collect reference trusts from several different service users to make better decisions. This can be called consensus trust. Assume a service consumer A needs to obtain the value of the trust in a criterion C of a service S, but he has no information about the trust of $S.C$. However, he has information about trust in other consumer X and Y, and both of they have a trust in $S.C$. Then the trust relationship between A and $S.C$ can be defined by :

$$A \rightarrow S.C = ((A \rightarrow X) \cap (X \rightarrow S.C)) \cup ((A \rightarrow Y) \cap (Y \rightarrow S.C)) \tag{8}$$

In this equation, the symbol $\cup$ means that A's trust in $S.C$ can be derived by combining X and Y's criterion trust in $S.C$.

Definition 3. ***Transitive Trust:*** *Suppose there are two service consumers A and B, where A has a reference trust in B. Additionally, B has a function trust in criterion C of service S. A's trust in P.S.C can be derived by using both B's function trust in S.C and A's trust in B:*

$$T(A \rightarrow P.S.C) = T(A \rightarrow B) \cdot T(B \rightarrow P.S.C) \tag{9}$$

Definition 4. ***Consensus Trust:*** *The consensus trust of two consumers' trust in P.S.C is a trust that reflects both trust in a fair and equal way. Then derived consensus trust in P.S.C, $T(A \rightarrow P.S.C)$, is calculated by:*

$$\frac{|T(A \rightarrow X) \cdot T(X \rightarrow P.S.C)| + |T(A \rightarrow Y) \cdot T(Y \rightarrow P.S.C)|}{|T(A \rightarrow X)| + |T(A \rightarrow Y)|} \tag{10}$$

IV. Criterion Value Estimation and Malicious Value Detection

Malicious service consumer can be classified as either adulating service consumer, which tries to falsely improve the trust in certain service providers, or defaming service consumer, trying to degrade the trust in certain service providers.

A. Criterion Value Estimation

It is reasonable to model the distribution of the value of $P.S.C$ as Normal distribution with (μ, σ), because the values of $P.S.C$ obtained by a consumer A are independent. For each criterion, its value follows normal distribution with $\{\mu^r, \sigma^r\}$, where μ^r is the real value of $P.S.C$'s μ, and σ^r is the actual $P.S.C$'s variance.

At one time, each service consumer i has an estimated value of $P.S.C$'s $\{\mu^r, \sigma^r\}$, denoted as $\{\mu^e_i, \sigma^e_i\}$, μ^e_i and σ^e_i represent the estimated real value of $P.S.C$ and estimated variance, respectively. A service consumer A is going to use estimated values of all other service consumers to predict $P.S.C$'s $\{\mu^r, \sigma^r\}$. After using service consumer i's estimated value, A's estimated values are denoted as $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$. Because of incomplete knowledge of the criterion of the service, i's estimated value usually has a deviation from A's estimated value $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$. Because the estimated value from many independent service consumers, the relation between i's estimate and A's estimate is modeled as

$$\begin{aligned} \mu^e_i = &\ \ \mu^e_{A,i} + \lambda_\mu \text{ and } p(\lambda_\mu) \sim \text{Normal}(0, \Lambda_\mu) \\ \sigma^e_i = &\ \ \sigma^e_{A,i} + \lambda_\sigma \text{ and } p(\lambda_\sigma) \sim \text{Normal}(0, \Lambda_\sigma) \end{aligned} \tag{11}$$

Note that λ_μ is different from σ^e_i. λ_μ is an estimate noise covariance when service consumer A estimating real value μ^r, while σ^e_i is estimated covariance from service consumer i, which may be malicious. Similarly, λ_σ is an estimate noise covariance when A estimating real value σ^r.

Based on Kalman Filter [16], the estimates of $\{\mu^r, \sigma^r\}$ are governed by the following linear stochastic difference equations:

$$\begin{aligned} \mu^e_{A,i} = F_\mu \mu^e_{A,i-1} + Bu_{i-1} + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu) \\ \sigma^e_{A,i} = F_\sigma \sigma^e_{A,i-1} + Bu_{i-1} + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma) \end{aligned} \tag{12}$$

where, F is the factor for relationship between the previous estimate based on the estimate from consumer $i-1$ and the current estimate based on i's estimate, and u is the optional control input to the estimate $\{\mu^e_A, \sigma^e_A\}$. Because in our model there is no control input, u is 0. Hence, our estimate is governed by the following linear difference equation:

$$\begin{aligned} \mu^e_{A,i} = F_\mu \mu^e_{A,i-1} + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu) \\ \sigma^e_{A,i} = F_\sigma \sigma^e_{A,i-1} + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma) \end{aligned} \tag{13}$$

In Kalman Filter, there are two steps: $Predict$ step and $Update$ step. P_μ and P_σ represents predict error covariance of $\mu^e_{A,i}$ and $\sigma^e_{A,i}$ respectively. By using $\{\mu^e_{A,i-1}, \sigma^e_{A,i-1}\}$, the $Predict$ step is responsible for obtaining the priori estimate, denoted by$\{\bar{\mu}^e_{A,i}, \bar{\sigma}^e_{A,i}\}$, for $Update$ step. Similarly, priori predict error covariances are denoted by $\bar{P}_\mu$ and $\bar{P}_\sigma$. The $Update$ step is responsible for incorporating a new service consumer's estimate $\{\mu^e_i, \sigma^e_i\}$ to obtain an improved posteriori estimate $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$.

$Predict$ step:

$$\bar{\mu}^e_{A,i} = F_{\mu,i} \mu^e_{A,i-1}, \quad \bar{\sigma}^e_{A,i} = F_{\sigma,i} \sigma^e_{A,i-1} \tag{14}$$

$$\bar{P}_{\mu,i} = F^2_{\mu,i} P_{\mu,i-1} + W_{\mu,i}, \quad \bar{P}_{\sigma,i} = F^2_{\sigma,i} P_{\sigma,i-1} + W_{\sigma,i} \tag{15}$$

$Update$ step:

$$K_{\mu,i} = \bar{P}_{\mu,i} / (\bar{P}_{\mu,i} + \Lambda_{\mu,i}), \quad K_{\sigma,i} = \bar{P}_{\sigma,i} / (\bar{P}_{\sigma,i} + \Lambda_{\sigma,i}) \tag{16}$$

$$\begin{aligned} \mu^e_{A,i} = &\ \ \bar{P}_{\mu,i} + K_{\mu,i}(\mu^e_i - \bar{\mu}^e_{A,i}), \\ \sigma^e_{A,i} = &\ \ \bar{P}_{\sigma,i} + K_{\sigma,i}(\sigma^e_i - \bar{\sigma}^e_{A,i}) \end{aligned} \tag{17}$$

$$P_{\mu,i} = (1 - K_{\mu,i}) \bar{P}_{\mu,i}, \quad P_{\sigma,i} = (1 - K_{\sigma,i}) \bar{P}_{\sigma,i} \tag{18}$$

In order to compute the parameters $F_{\mu,i}$, $\Lambda_{\mu,i}$, $W_{\mu,i}$, $F_{\sigma,i}$, $\Lambda_{\sigma,i}$, $W_{\sigma,i}$, the following equations are used:

$$F_{\mu,i} = \frac{\sum_{j=1}^{i-1} \mu^e_{A,j} \mu^e_{A,j-1}}{\sum_{j=1}^{i-1} (\mu^e_{A,j})^2}, \quad F_{\sigma,i} = \frac{\sum_{j=1}^{i-1} \sigma^e_{A,j} \sigma^e_{A,j-1}}{\sum_{j=1}^{i-1} (\sigma^e_{A,j})^2} \tag{19}$$

$$\Lambda_{\mu,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\mu^e_j - \mu^e_{A,j})^2, \quad \Lambda_{\sigma,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\sigma^e_j - \sigma^e_{A,j})^2 \tag{20}$$

$$\begin{aligned} W_{\mu,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\mu^e_{A,j} - F_i \mu^e_{A,j-1})^2, \\ W_{\sigma,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\sigma^e_{A,j} - F_i \sigma^e_{A,j-1})^2 \end{aligned} \tag{21}$$

B. Malicious Value Detection

Given significance probability levels δ_μ and δ_σ, the problem of determine if the service consumer i is not malicious is to find the threshold values $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ so that:

$$P(|\mu^e_i - \mu^e_{A,i}| \leq \Delta_{\mu,i}) = \delta_\mu, P(|\sigma^e_i - \sigma^e_{A,i}| \leq \Delta_{\sigma,i}) = \delta_\sigma \tag{22}$$

In addition, $\mu^e_i - \mu^e_{A,i}$ and $\sigma^e_i - \sigma^e_{A,i}$ follow zero mean normal distribution with variance $P_{\mu,i} + \Lambda_{\mu,i}$ and $P_{\sigma,i} + \Lambda_{\sigma,i}$ respectively. Hence, there are also equations:

$$\begin{aligned} P(|\mu^e_i - \mu^e_{A,i}| \leq \Delta_{\mu,i}) = 1 - 2\Phi\left(\frac{-\Delta_{\mu,i}}{\sqrt{P_{\mu,i} + \Lambda_{\mu,i}}}\right), \\ P(|\sigma^e_i - \sigma^e_{A,i}| \leq \Delta_{\sigma,i}) = 1 - 2\Phi\left(\frac{-\Delta_{\sigma,i}}{\sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}}}\right) \end{aligned} \tag{23}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Hence, after solving Equations. (22) and (23), $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ can be obtained:

$$\begin{aligned} \Delta_{\mu,i} = -\Phi^{-1}((1 - \delta_\mu)/2) \sqrt{P_{\mu,i} + \Lambda_{\mu,i}}, \\ \Delta_{\sigma,i} = -\Phi^{-1}((1 - \delta_\sigma)/2) \sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}} \end{aligned} \tag{24}$$

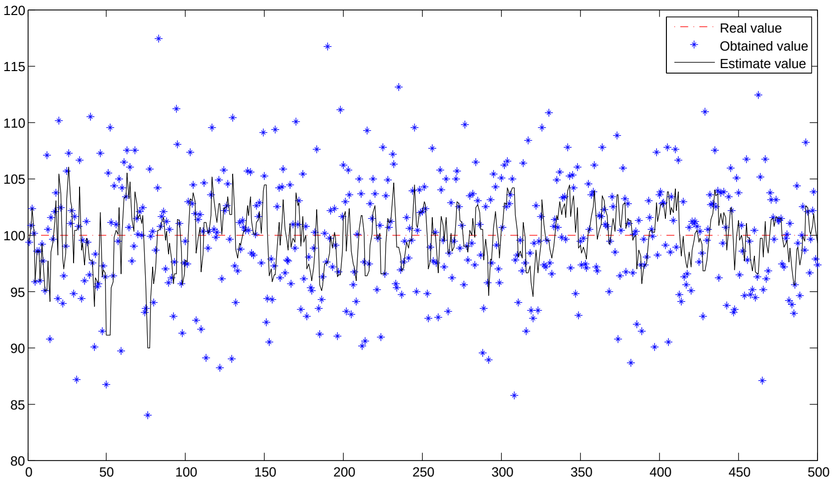


Fig. 1. The real, estimate and obtained values

C. Calculation Algorithm

Definition 5. ***Time Domain****: Each time a service consumer invokes a service, it can obtain values of all criteria. Then it can use the values to estimate real criterion value and the values are called values of quality criterion at Time Domain.*

Because at time domain all values are collected by a service consumer A itself, it is unnecessary to detect malicious values. For ratio criteria, it is easy to calculate the values and it is accurate. For instance, the success rate c_{sr} of a service S can be calculated by the following equations:

If the i^{th} invocation of service S is successful:

$$\text{num}_{\text{success},i} = \text{num}_{\text{success},i-1} + 1 \tag{25}$$

If the i^{th} invocation of service S fails:

$$\text{num}_{\text{success},i} = \text{num}_{\text{success},i-1} \tag{26}$$

Finally:

$$\text{num}_{\text{total},i} = \text{num}_{\text{total},i-1} + 1 \tag{27}$$

$$c_{sr} = \frac{\text{num}_{\text{success},i}}{\text{num}_{\text{total},i}} \tag{28}$$

Each time a service S invoked by a service consumer i, then i can get values of non-ratio criteria. If exact values of non-ratio criteria C can be obtained, such as price, then the value $\{\mu_i^e, \sigma_i^e\}$ can also be calculated by the equations as follows:

$$\mu_i^e = E[C_i] \tag{29}$$

$$\sigma_i^e = E[(C_i - \mu_i^e)^2] \tag{30}$$

However, exact values of certain non-ratio criteria cannot be obtained, such as response time. Not only because computer is a complex dynamic system, but also because of network delay, it is impossible to get exact response time of a service. Hence, at this point, the method of criterion value estimation in Section IV is used to obtain the estimate value $\{\mu_i^e, \sigma_i^e\}$.

Definition 6. ***Aggregation Domain****: Each service consumer can collect criterion values of various services from numerous service consumers. Further these values can be aggregated by this service consumer to estimate real criterion value too, although some of these values may be malicious. These values are called values of quality criterion at Aggregation Domain.*

At aggregation domain, values of all criteria including ratio criteria and non-ratio criteria are estimated by using the method of criterion value estimation in Section IV, not only because malicious values need to be filtered out, but also because all these values may not be accurate due to incomplete knowledge on service providers.

Assume a consumer A is going to aggregate all values from others to calculate the trust in a quality criterion C of a service S provided by service provider P. Then each step after estimating the value of C by the use of the value provided by other service consumer X, the trust in C is calculated using the method presented in Section III-A. When A use this value to estimate the trust in $P.S.C$, the trust in other consumer X calculated by the use of the method introduced in Section III-B is used to weight the trust in a service provider from that consumer X . Furthermore, second hand values are also aggregated using the approach presented in Section III-C.

V. Performance Evaluation

In this section, this trust model is evaluated in a simulated environment. The model is validated first, and then another experiment is carried out to evaluate the robustness of this model, compared to some other approaches.

The first experiment is carried out in a clean environment without malicious values in order to validate the model. The value of a QoS criterion C is estimated and the accuracy of the value estimation is evaluated each step. One result calculated by our model is shown in Fig.1. The dashed line represents C's real values, while the obtained value with noise is denoted by the stars. The real line represents the values estimated by our model. It is seen that in this experiment the obtained values

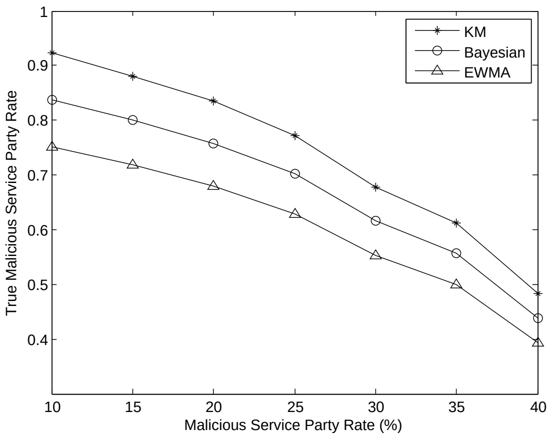


Fig. 2. Average true malicious rate

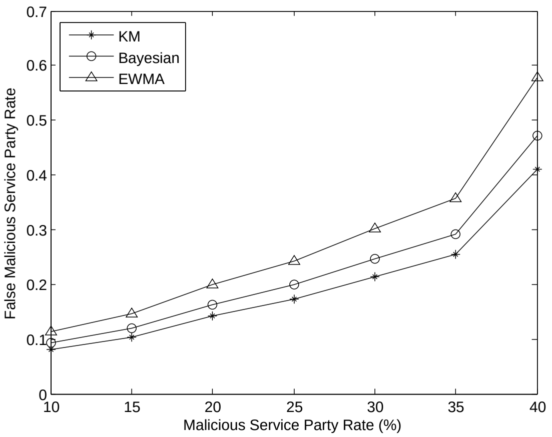


Fig. 3. Average false malicious rate

are not accurate, but C's values are still being estimated well, which is closer to the real values than the obtained values.

In order to evaluate the robustness of this model, another experiment is carried out in an environment with malicious values. Our approach is compared with the methods in [9], [13], respectively represented by EWMA and Bayesian, to evaluate the robustness of malicious value detection. Because it is almost impossible that more than a half of all service consumers within an environment behaviour maliciously and it is impossible to perform well in an environment with more than 50% malicious service consumers, the probability of malicious service consumers is set up to 40%.

Definition 7. ***True Malicious Rate:*** *The percentage of correctly detected malicious service consumers.*

The number of malicious values and correctly detected malicious ones are denoted by num_m and num_c respectively, and true malicious rate is calculated by $\dfrac{\mathrm{num}_c}{\mathrm{num}_m}$.

Definition 8. ***False Malicious Rate:*** *The percentage of wrongly detected non-malicious service consumers.*

The number of all non-malicious and wrongly detected malicious values are denoted by $\mathrm{num}_{\mathrm{non}}$ and num_w respectively, and then false malicious rate is calculated by $\dfrac{\mathrm{num}_w}{\mathrm{num}_{\mathrm{non}}}$.

As shown in Figs. 2 and 3, as the increase of the malicious service consumer probability, our model performs better than those two approaches. Hence, the accuracy of those approaches is lower than ours.

VI. Conclusion

Trust in Quality of Service of service providers is really important for service consumers to select services. In this paper, a model using Kalman Filter to filter out malicious values was introduced. First, QoS criteria were classified into several groups on the basis of their characteristics. Then a model to estimate the trust in quality criterion was presented, not only based on the trust in service providers but also on the basis of the trust in service consumers, which significantly helped reduce the influence of dishonest service consumers. The trust calculation processes were classified into two groups, including Time and Aggregation Domain. At time domain, a service consumer uses the values obtained by itself while at aggregation domain, a service consumer to calculate the value of trust in a service provider by the use of values from other service consumers, which may be malicious. Hence, at aggregation domain, a method based on Kalman Filter was presented to filter out malicious values and the trust in other consumer X was used to weight the data from X. Finally, our model was evaluated by two experiments and the results shown that a more accurate value estimation can be made, with better detection accuracy, compared with two other approaches.

Although our model works well, a large amount of historic estimation needs to be stored and it needs lots of calculation. Hence, further research will be carried out to reduce the need of storing historic estimation and calculation.

Acknowledgment

This work is partly supported by British Telecommunications Research. The authors would like to thank Keith Briggs, Michael Fitch, Santosh Kawade, etc.

References

[1] S. R. Yan, X. L. Zheng, D. R. Chen, and W. Y. Zhang, "User-centric trust and reputation model for personal and trusted service selection," International Journal of Intelligent Systems, vol. 26, no. 8, 2011, pp. 687–717.

[2] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in Proceedings of the 12th International Conference on World Wide Web, 2003, pp. 640–651.

[3] Z. Yan and H. Silke, "Trust modeling and management: From social trust to digital trust," in Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions, 2008, pp. 290–323.

[4] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," ACM Compute Survey, vol. 42, no. 1, 2009, pp. 1:1–1:31.

[5] Y. Wang and K.-J. Lin, "Reputation-oriented trustworthy computing in e-commerce environments," Internet Computing, IEEE, vol. 12, no. 4, 2008, pp. 55–59.

[6] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support System, vol. 43, no. 2, 2007, pp. 618–644.

[7] D. Ardagna and B. Pernici, “Adaptive service composition in flexible processes,” Software Engineering, IEEE Transactions on, vol. 33, no. 6, 2007, pp. 369–384.
[8] Y. H. Wang and M. P. Singh, “Trust Representation and Aggregation in a Distributed Agent System,” in Proceeding of National Conference on Artificial Intelligence, 2006, pp. 1425–1430.
[9] Y. C. Zhang and Y. G. Fang, “A fine-grained reputation system for reliable service selection in peer-to-peer networks,” Parallel and Distributed Systems, IEEE Transactions on, vol. 18, no. 8, 2007, pp. 1134–1145.
[10] A. Whitby, A. Josang, and J. Indulska, “Filtering out unfair ratings in bayesian reputation systems,” in Proceeding of the International Joint Conference on Autonomous Agenst Systems, 2004, pp. 106–117.
[11] S. Buchegger and J.-Y. Le Boudec, “A robust reputation system for peer-to-peer and mobile ad-hoc networks,” in Proceedings of the Second Workshop Economics of P2P Systems, 2004, pp. 1–6.
[12] X. F. Wang, L. Liu, and J. S. Su, “Rlm: A general model for trust representation and aggregation,” Services Computing, IEEE Transactions on, vol. 5, no. 1, 2012, pp. 131–143.
[13] W. Conner, I. Rouvellou, A. Iyengar, K. Nahrstedt, and T. Mikalsen, “A trust management framework for service-oriented environments,” in International World Wide Web Conference, 2009, pp. 891–900.
[14] L. Xiong and L. Liu, “Peertrust: supporting reputation-based trust for peer-to-peer electronic communities,” Knowledge and Data Engineering, IEEE Transactions on, vol. 16, no. 7, 2004, pp. 843 – 857.
[15] M. Srivatsa, L. Xiong, and L. Liu, “Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks,” in Proceedings of the 14th international conference on World Wide Web, 2005, pp. 422–431.
[16] R. E. Kalman “A New Approach to Linear Filtering and Prediction Problems,” Transaction of the ASME-Journal of Basic Engineering, vol. 82, 1960, pp. 35-45.

Reputation as a Service

João C. Ferreira, Porfírio P. Filipe, and Paulo M. Martins
Instituto Superior de Engenharia de Lisboa
Lisbon, Portugal
{jferreira@deetc, pfilipe@deetc, paulo.martins@dec}.isel.pt

Abstract— **In this research work, we introduce the concept of a reputation service to evaluate user collaboration towards community or system goal. Online reputation mechanisms allow members of a community to submit their opinions (feedback) regarding other community members and their publication activity. Submitted opinions are analyzed, aggregated with feedback posted by other members and made publicly available to the community in the form of member feedback profiles. We propose a conceptual system that can be used in several contexts, namely in our public transportation recommender system developed in the framework of the European Project START.**

Keywords-Reputation; Recomemnder System; Public Transportation; Service

I. INTRODUCTION

Online reputation mechanisms allow members of a community to submit their opinions (feedback) regarding other community members. Submitted opinions are analyzed, aggregated with feedback posted by other members and made publicly available to the community in the form of member feedback profiles. Several examples of such mechanisms can already be found in a number of diverse online communities. Perhaps the best-known application of online reputation mechanisms are the trust system in electronic markets. The first Web sites to introduce reputation schemes were on-line auction sites such as eBay.com. They are now also used by company reputation rating sites such as BizRate.com, which ranks merchants on the basis of customer ratings. Consumer Reports Online's eRatings rates merchants on the basis of test purchases carried out by Consumer Reports staff.

Online reputation systems have appeared as a viable instrument for encouraging cooperation among strangers in such settings by guaranteeing that the behavior of a player towards any other player becomes publicly known and may therefore affect the behavior of the entire community towards that player in the future. Knowing this, players have an incentive to behave well towards each other, even if their relationship is a one-time deal. The rise of online reputation systems is already indirectly changing people's behavior. Circumstantial evidence suggests that people now increasingly rely on opinions posted on such systems in order to make a variety of decisions. The growing importance of online reputation systems in different subject raise new challengers in the study of participant's behavior in the communities where they are introduced. We have developed several recommender systems for Electric vehicles [1], transportation system [2], electric market [3] and the study of user behavior was always created. This raise the question of a definition of a service that can be used for several systems (recommender or not) in order to evaluate user performance towards the system or community goal.

This work is divided eight sections. In Section I, we define the problem context. Section II describes the European project associated with our work. Section III describes the recommender engine used to send user relevant information. In Section IV, we present the collaboration system used by the registered users to create and share transportation related information. Section V presents the reputation system to motivate and evaluate user participation towards a common system goal. In Section VI, we describe the developed approach based on services. Section VII presents the system (Social Mobility Advisor) developed. Finally, conclusions are drawn in Section VIII.

II. START PROJECT

This work describes ISEL (Instituto Superior Engenharia de Lisboa, Portuguese polytechnic) contribution of a reputation system applied for a public transportation recommender system for the Seamless Travel across the START project (Atlantic Regions using sustainable Transport). START Project is a European Commission's Transnational Territorial Cooperation Program with 14 partners from the UK, France, Spain and Portugal. Its mission is the establishing of a transnational network of regional & local authorities to promote enhanced accessibility, giving tools to make easy to travel to, from, and around the Atlantic regions using environmentally friendly, collective modes of transport greater interconnectivity between transport systems clearer information within regional gateways, airport hubs ports & rail interchanges. The focus on this work is aligned with the "Integra concept" [4], whose aim is to provide a single brand that links together and provides information on the different public transport operations across the Atlantic Area. So, the system should allow the query of multiple information sources through a unique interface. The queries and answers to them should reflect a single data model. The existence of this common data model takes the software applications with the difficult task of dealing with various technologies and their relational schemas. Different public transportation systems can be added from the end user point of view. Also, this integration allows the creation of mobile systems oriented for tourism purposes, other main goal of Integra, where “low budget tourism” can be guided, to reach POI (Point of Interest) by public transportation.

III. Recommender Engine

The Recommender Engine (RE) is the same engine used for the recommendation of electric vehicles defined in [1]. This engine matches information dedicated to public transportation from web and our social network with the user profile to bring the right information to a mobile device. Applied to the reality of smart cities with increasing mobility and sustainability needs the proposed system integrates a diversity of functionalities. The main system modules are: (1) GPS module: The RE receives the information of the geographical positioning information on the current position of the vehicle and the features that enable the calculation of distances between two points; (2) Public transportation Information module: a system with public transportation routes, schedules. The reception of such information is through an exchange file in XML format that contains information prepared and compiled by different operators. The file includes the geographical location of sites where you can embark on public transport scheme. The public transport information is incorporated into items of candidate recommendation system by allocating an item property that indicates whether or not is close to public transport. This information is another dimension that enters the calculation of the usefulness of each item positively or negatively affects your score accordingly to the choices that have been made before by the driver; (3) collaboration system to handle user participation towards community goal.

IV. Collaboration System

Users can also interact and collaborate among themselves to improve their knowledge, or by allowing them to express their needs, preferences and also create valuable information related to public transportation. The features presented below are created to keep users informed, motivated and with intention to collaborate more frequently:

- Cooperation Area: An area where users can ask or provide different kinds of knowledge to cooperate in Public transportation information;
- Helping Area: An area where users can post questions and answers and some sort of help of any topic. This space could be accessed and viewed by any registered user. The system gives credits to users that provide good helping answers to posted questions, based on the following modules;
- Abuses or Faults Reporting Area: An area provided to report abuses of different kinds, like comments or bad use of the System. The system manager can penalize the users for inappropriate behavior. The reason for providing this area is essentially for discouraging users to commit abuses or faults;
- Request: An area where users can ask for specific questions. The system manager can use those requests to tune the community;
- Reward: Created in order to promote and recognize outstanding behavior (for example, no changes in the profile). The awarded users increase their reputation;
- Community Newsletter: The system Manager publishes a digital newsletter with Community information and Public transportation related information;
- Users Reputation: User reputation represents the most valuable collaborators, see reputation system;
- Alerts Subscription: Users can subscribe to different kinds of alerts: notifications, comments or other Public Transportation interactions.

V. Reputation System

A reputation system collects, distributes, and aggregates feedback about participants’ past behavior, helping users to decide whom to trust and encouraging trustworthy behavior [5]. There are many empirical studies reputation system, see Resnick et al. [6] for an overview. Various trust and reputation systems have been proposed or implemented in different areas. Some proposed classifications use their application areas [7], algorithms [5] or some combined characteristics [8] as criteria. These classifications allow looking at trust and reputation systems from different perspectives always from zero development phases. Since there are common modules and strategies among different systems, the idea is to build one based on a modular structure using a service approach.

The Reputation system implements a model that is generally composed of the main functions:

- AddPoints/SubPoints – Add or remove user points based on pre-definied criteria;
- DemoteUser/PromoteUser – Change user reputation level based on points criteria;
- GetReputationToLevel – input is the current user reputation and gives the points necessary to go to the next level;
- Rate/Vote – user item evaluation for other user annotation and gives as output the new rate for that annotation item.

Several reputation algorithms can be implemented. At this moment, we implemented binary rate and the start algorithm.

A. Reputation Level

The reputation levels system is inspired from videogames like World of Warcraft [9], which needs exponential requirements to level up. Equation 1 illustrates the adopted formula to generate reputation level requirements for the next level, supplying the current level.

$$PN = (8 * NX) * (45 + (5 * NX) \qquad (1)$$

PN is the reputation for next user level

NX is the reputation for current user level

ISBN: 978-1-61208-270-7

So, from the first to the second level, the user needs 400 points, from the second to the third level, 880 points, from the third to the fourth level, 1440 points, and from the fourth to the fifth level, 2080 points. The reputation level is an important issue in our reputation system, since it will define the weight of the user's ratings, described below, as well as their role/privileges in Social Network.

B. *Ratings*

Our system provides an input for users to express their opinion related to item's information quality by rating it as being helpful or not helpful (+1, -1), as shown in Fig. 1. This range permits a less ambiguous evaluation, since it will influence the reputation of those who submitted the item. Using this schema, it is possible to filter and organize information by its quality and, at the same time, indirectly rate the user who submitted that information. This approach aims to decrease public reputation of those who submit poor information and, inversely, reward those who submit relevant information. Item's information reputation is determined using the Equation 2, which consists on a summation of all positive and negative ratings weighted by rater's reputation level. Rater's reputation level is used to avoid unfair ratings [1], one of reputation systems known problem, in assumption that users with better reputation provide more reliable ratings.

$A_i = a * b - (c' * d') + c * d$ (2)

a – Item i current rating (in cache);
b – Summation of item i raters reputation (in cache);
c – New rating for item i of active user;
d – Item i active user reputation level (in cache);
Reevaluation factors:
c' – Old rating for item i of active user;
d' – Reputation level of active user when submitted the old evaluation (in cache);

In order to calculate the mobility item's quality itself (e.g., how good is a gate, operator, transport, etc.), our system also provides an input for users to express their mobility experience quality by a quantitative rating from 1 to 5 stars, plus comment.

The item's quality reputation is calculated by the Equation 3 and it consists in a weighted average between evaluations and raters reputation. Rater's reputation level is used to avoid unfair ratings as mentioned before.

$A_i = (a * b - (c' * d') + c * d) / (b - d' + d)$ (3)

a – Item i current rating (in cache);
b – Summation of item i raters reputation (in cache);
c – New rating for item i of active user;
d – Item i active user reputation level (in cache);
Reevaluation factors:
c' – Old rating for item i of active user;
d' – Reputation level of active user when submitted the old evaluation (in cache);

C. *User's reputation*

As mentioned before, users indirectly rate others by rating their submitted items as being helpful or not helpful, according to the Equation 4:

$R = - (a' * r') + a * r$ (4)

R – Reputation increment; a – Rater reputation level;
r – New rating (+1, -1);
Reevaluation factors:
a' – Old rater reputation level; r' – Old rating;

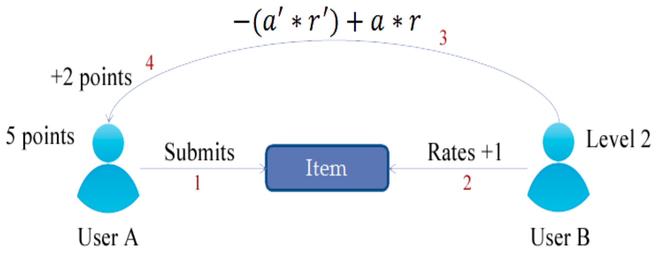


Figure 1. Evaluation example.

For example, when a user A submits an item and a second user B on level 2 rates that item as useful (+1), the user A gets two points of reputation as presented in Fig. 1.

D. *Interaction Incentives*

Due to the lack of incentive for users to provide their mobility experiences and ratings, we included incentive mechanisms in our reputation system. Users are rewarded for data submission, where the importance of the submitted item defines the assigned reputation points. Table 1 shows the possible configuration values of rewarded points.

TABLE 1: POSSIBLE CONFIGURATION FOR INCENTIVES.

Action	Reputation
Binary rating	+2
Start rating	+1
Mobility item submission, except route	+4
Route submission	+6
Comment submission	+1
Collaboration action	+3
Reevaluation	0

For example, someone that submits a new gate will be rewarded with 4 reputation points.

We also adopted a badge reward system, used by systems like Stackoverflow [10] and Foursquare [11] with great acceptance by the community, where users are rewarded by reaching certain objectives.

In our context, we reward the users when they reach, for example, a pre-defined number of done routes or rated items.

E. *Moderation*

In our approach, the information is strongly dependent on user's collaborative interactions. Consequently, moderation actions are needed. To reduce the amount of necessary staff to keep the system, we included a role based moderation mechanism, inspired by Slashdot, which takes advantage of user's reputation to delegate them moderation actions. This way, most reliable users are assigned to perform some moderation actions. Our mechanism has two moderation roles, where the first, called "collaborator", are assigned to manage regular users by editing/hiding comments and item's information. The second role, called "moderator", was introduced to reduce the number of unfair collaborators, as described by Slashdot [5].

VI. SOFTWARE DEVELOPMENT

The concept normalization in public transportation area is achieved by the introduction of domain ontology. We use same recommender process to several problems in the transportation domain, such as a recommender engine for electric vehicle [1] and current proposal.

Ontologies are very powerful in the sense that they are developed with the human understanding of the domain in mind, instead of taking a pure application-oriented approach, as it is mostly done with database schemas. Ontologies can help bridge the gap between human understanding and machine understanding of a domain. We developed a domain ontology based on Web Ontology Language (OWL) and UML2 profile and a mapping process between different database schemas to a central Information Model or to other database schemas, and in that way enable better understanding of this information by the users in the organization. Public Transportation ontology (PTO) can play an important role providing mechanisms to handle automatically data integration among different Public transportation data and all related data acquired with user participation. PTO is described in [2]. PTO is built on top of RDF, thus it inherits its concepts: resource, propriety, datatype and class: (1) resource is one of the bases of RDF, it represents all things described and is the root construction. It is an instance of MOF classes; (2) property, defines the relation between subjects and object resources and is used to represent relationships between concepts. Ontology class attributes or associations are represented through proprieties; (3) ontology is a concept that aggregates other concepts (classes, properties, etc). It groups instances of other concepts that represent similar or related knowledge; (4) classifier is the base class of concepts that are used for classification and is divided in: (i) datatype, a mechanism for grouping primitive data; (ii) abstract class; and (5) instance that is the base class and is divided in individuals and data values, for details see [2].

Taking into account a recommender system to be developed, in our approach, the system developer can use reputation service and chose the available reputation algorithms. To obtain that, it is necessary to use a Service-Oriented Architecture (SOA) [12], which is a component-based software architecture. This architecture defines the description of the services' interface and implementation. The interface is defined in an unbiased way, independent of the hardware platform, operating system and programming language of the services implementation. For implementing service request software architecture, SOA changes the manner of the traditional software development. The usage of Web Services, the combination of the related technologies and the software architecture above are the basis for the implementation of the Web Services. This Web Services model is divided into three main parts: (1) Services Interface; (2) Services Mapping; and (3) Services Implementation. Services in SOA also include Services Contract, and Service Contract is the definition of the Services interface and Services Implementation. Services Interface in this model is realized by Web Services Definition Language (WSDL) [14], following the criterion of the WSDL. Services implementation follows the criterion of the special platform. Because of the universal property of WSDL, the services in this model own the universal property.

Representational State Transfer (REST) [15], describes a series of architectural constraints that exemplify how the web's design emerged. Several implementations of these ideas have been created throughout time. As a set of design criteria, REST is very general. In particular, it is not tied to the Web. Nothing about REST depends on the mechanics of HTTP or the structure of URIs. So, Web Service is designed according to resource. Every Web Service represents a kind of resource, which is operated by the operations of Web Service. The Resource Oriented Architecture (ROA) is a way of turning a problem into a RESTful [16] web service: an arrangement of URIs, HTTP, and XML that works like the rest of the Web.

A new, rational structure, high consistency, low connection of SOA-based application architecture is constructed, which also follows ROA, implements services by encapsulation of the entity object and its operations using Web Services, implements the business processes in the client PC. Compared with the traditional software architecture, the new services layer is added between the entity object layer and business logic layer. In the new layer, data and its operations are packaged to Web Services. The implementation of services is based on the "Interface-First" principle, for sharing the data fully and safely. In the entity object layer, based on the OOP (Object-Oriented Principle), the entity objects and their operations are delivered to three parts: (1) Entity Class; (2) Interface of Data Access; and (3) Data Access [17]. An overview of this process is illustrated in Fig. 2.

According to ROA, Web Service is designed around the resource. To simplify the development, the extension of the resource's definition is reduced, and the table of the database is a basic kind of resource. In the period of object oriented development, O/R mapping (Object/Relational mapping) is adopted to resolve the problem of Object-relational impedance mismatch [10]. This strategy combines business logic and data access for stripping. It increases the performance of the business logic for the persistence of the interaction between objects, instead of directly operating on the database tables and fields.

Web Service is used to disjoin the business logic and the operation of persistent object. The simple operation of the persistent object is packaged to Web Service, to share the resource commonly and easily.

According to the basic operations of the table, the operations of the Web Service are designed as list: (1) Add object to the database; (2) Delete object from the database;

(3) Edit the object, and save the result to the database; (4) Find object in the database; (5) Find a list of objects in the database.

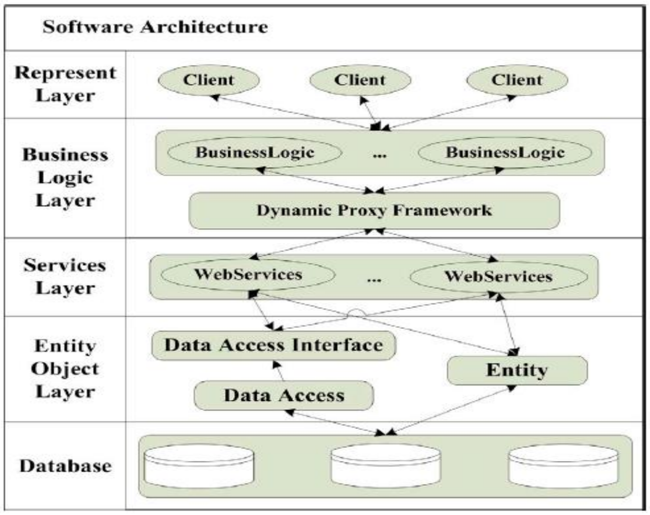


Figure 2. Software Architecture Based on SOA [16].

VII. Social Mobility Advisor

Social Mobility Advisor (SMA), was our Public Transportation Recommender system developed for START project using a service oriented approach. The main services introduced are: (1) geo-location, based on HTML 5 – API, from the user IP address, is possible to know the approximate location; (2) historical navigation data based on Address.js [17]; (3) Public transportation data were imported based on a semantic approach developed [2]; (4) external information from Google Maps and Google Places (maps and points of interests [19]), Brighter Planet for CO_2 calculation of carbon footprint taking into account trip start and end point [20], Wikipedia to get information not available in internal data base [21] and Facebook to get information about users (name, friends, e-mail, photo); (5) user authentication LDAP; (6) user reputation; (7) matching algorithm; and (8) user profile.

SMA main menu, illustrated in Fig. 3, is a collaborative solution that aims to assist travelers to share their experiences and find the best suitable mobility alternatives, bearing in mind the best sustainable options. Those actions are supported by the collaborative community interactions, in other words, under the social network concept. This approach leads to a credibility problem, which we took into account by assuring both users and information are cataloged by confidence levels. We also propose an automated moderation mechanism that tries to reduce the number of required staff to administer the system.

The SMA project aims to contribute to a more accessible, social and sustainable mobility across the Europe, dealing with reputation and recommender systems in order to build a social network.

VIII. Conclusions

The Integra social network, built from the experience of the SMA Project, has exceeded the testing phase, which counted with more than two hundred users. The present database includes more than ten thousand items (gates, transports, operators), covering mobility information in a worldwide context, and about one thousand rates and comments. This recommender system for public transportation was developed in a service based approach. Everyone can obtain more information using [22] and join the Integra social network accessing [23] and a full implementation description available at [24].

References

[1] J. C. Ferreira, P. Pereira, P. Filipe, and J. Afonso. “Recommender System for Drivers of Electric Vehicles,” Proc. IEEE 3rd International Conference on Electronics Computer Technology (ICECT 2011), 8-10 April 2011, pp. 244-248, Kanyakumari, India.

[2] J. C. Ferreira, P. Filipe, and A. Silva. “Multi-Modal Transportation Advisor System,” Proc. First IEEE FISTS Forum on June 2011, pp 388-393 in Vienna Austria.

[3] J. C. Ferreira, A. Silva, V. Monteiro, and J. L. Afonso. “Collaborative Broker for Distributed Energy Resources,” Computational Intelligence and Decision Making - Trends and Applications, Springer, 2013, in press.

[4] Integra Concept - http://www.start-project.eu/en/Integra.aspx, [retrieved: April, 2013].

[5] A. Jøsang, R. Ismail, and C. Boyd. “A Survey of Trust and Reputation Systems for Online Service Provision,” Decision Support Systems, pp 618-644, Volume 43, Issue 2, March 2007, ISSN 0167-9236.

[6] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment. Experimental Economics, 2006, pp. 79-101. Avaliable from http://www.si.umich.edu/~presnick/papers/postcards/PostcardsFinalPrePub.pdf, [retrieved: April, 2013].

[7] L. Mui, A. Halberstadt, and M. Mohtashemi. “Notions of reputation in multi-agents systems: A review,” Proc. Autonomous Agents & Multiagent Systems, 2002, pp. 280–287, Bologna, Italy.

[8] J. Sabater. “Trust and Reputation for Agent Societies,” PhD thesis, Institute for Artificial Intelligence Research, Bellaterra, 2003.

[9] World of Warcraft - http://eu.battle.net, [retrieved: April, 2013].

[10] Starckoverflow web page - http://www.stackoverflow.com, [retrieved: April, 2013].

[11] Foursquare web page - https://www.foursquare.com, [retrieved: April, 2013].

[12] D. Krafzig, K. Banke, and D. K. Slama. “Enterprise SOA: Service-Oriented Architecture Best Practices,” PrenticeHall, America, 2006.

[13] IBM DW. “Specification of the Architecture based on SOA,” , [retrieved: April, 2013].

[14] http://tech.ccidnet.com/pub/article/c322-a206969-p2.html, [retrieved: April, 2013].

[15] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. “Web Services Description Language (WSDL) 1.1,” http://www.w3.org/TR/wsdl, 2001, [retrieved: April, 2013].

[16] R. T. Fielding. “Architectural Styles and the Design of Network-based Software Architectures,” Doctoral Thesis, University of California, Irvine, America, 2000.

[17] L. Richardson and S. Ruby. “RESTful Web Services,” O'Reilly Media Inc, America, 2007.

[18] G. Xiao-Feng, Y. Shi-Jun, and Y. Zu-Wei. “Design and implementation of framework of web application based on .NET,” Computer Engineer and Design, Beijing, Vol. 29, No. 2, 2008.

[19] http://www.asual.com/jquery/address, [retrieved: April, 2013].

[20] Google Maps API Family - http://code.google.com/intl/pt-PT/apis/maps/index.html, [retrieved: April, 2013].

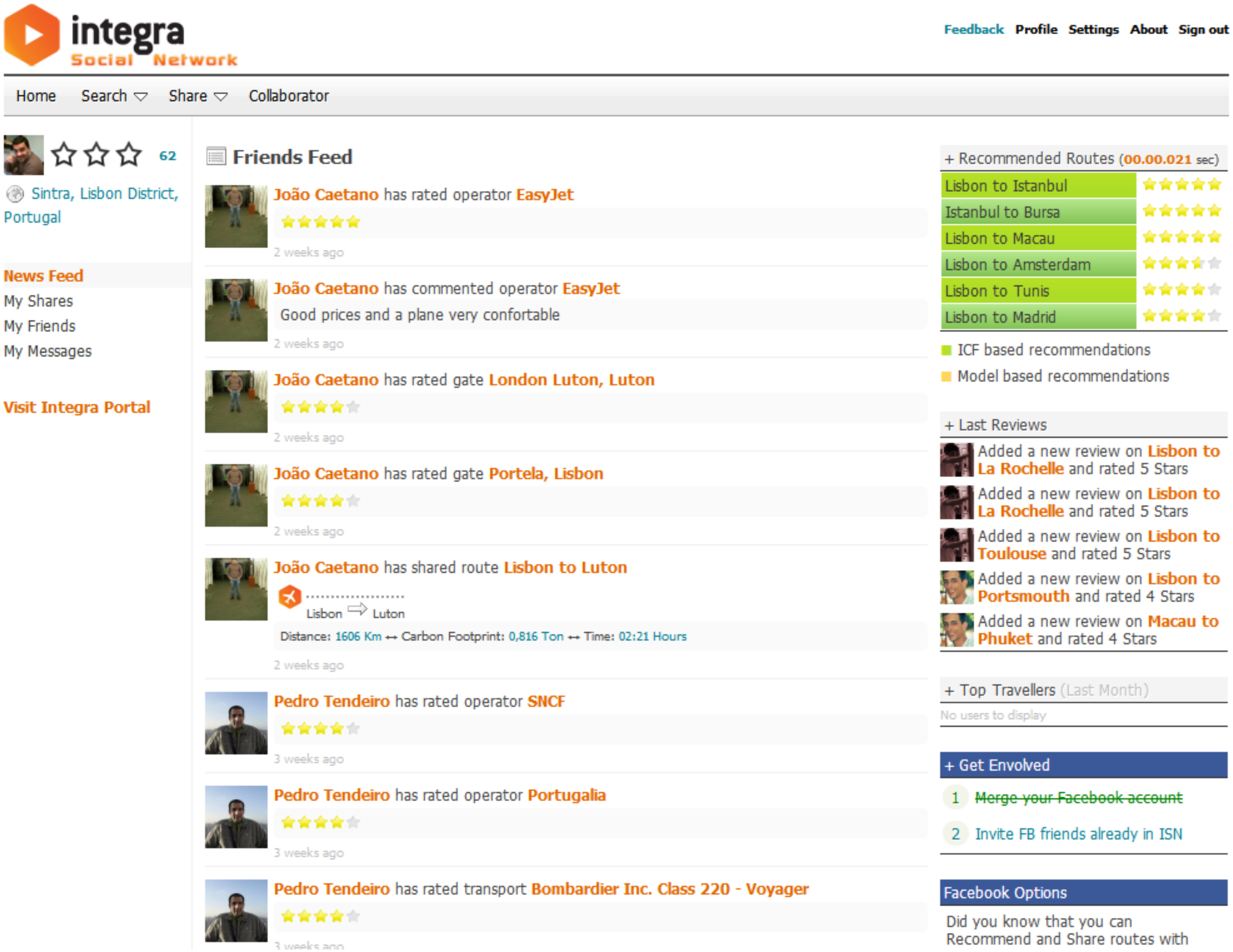
[21] Brighter Planet - http://brighterplanet.com, [retrieved: April, 2013].

[22] Media Wiki API - http://www.mediawiki.org/wiki/API, [retrieved: April, 2013].

[23] http://start.isel.pt, [retrieved: April, 2013].

[24] http://integra.isel.pt, [retrieved: April, 2013].

[25] Social Mobility Advisor, available at http://start.isel.pt/publications.html, [retrieved: April, 2013].



Friends Feed

João Caetano has rated operator EasyJet
★★★★★
2 weeks ago

João Caetano has commented operator EasyJet
Good prices and a plane very confortable
2 weeks ago

João Caetano has rated gate London Luton, Luton
★★★★☆
2 weeks ago

João Caetano has rated gate Portela, Lisbon
★★★★☆
2 weeks ago

João Caetano has shared route Lisbon to Luton
Lisbon ⇨ Luton
Distance: 1606 Km ↔ Carbon Footprint: 0,816 Ton ↔ Time: 02:21 Hours
2 weeks ago

Pedro Tendeiro has rated operator SNCF
★★★★☆
3 weeks ago

Pedro Tendeiro has rated operator Portugalia
★★★★☆
3 weeks ago

Pedro Tendeiro has rated transport Bombardier Inc. Class 220 - Voyager
★★★★☆
3 weeks ago

+ Recommended Routes (00.00.021 sec)

Lisbon to Istanbul	★★★★★
Istanbul to Bursa	★★★★★
Lisbon to Macau	★★★★★
Lisbon to Amsterdam	★★★½☆
Lisbon to Tunis	★★★★☆
Lisbon to Madrid	★★★★☆

🟩 ICF based recommendations
🟨 Model based recommendations

+ Last Reviews

Added a new review on Lisbon to La Rochelle and rated 5 Stars

Added a new review on Lisbon to La Rochelle and rated 5 Stars

Added a new review on Lisbon to Toulouse and rated 5 Stars

Added a new review on Lisbon to Portsmouth and rated 4 Stars

Added a new review on Macau to Phuket and rated 4 Stars

+ Top Travellers (Last Month)
No users to display

+ Get Envolved
1 ~~Merge your Facebook account~~
2 Invite FB friends already in ISN

Facebook Options
Did you know that you can Recommend and Share routes with

Figure 3. SMA application screen (Integra Social Network).

Privacy Monitoring and Assessment for Ubiquitous Systems

Architecture design and Java prototype implementation

Mitja Vardjan and Jan Porekar
Research department
SETCCE
Ljubljana, Slovenia
{mitja.vardjan, jan.porekar}@setcce.si

***Abstract*—Pervasive services and their respective front-ends can have access to large amounts of personally identifying and sensitive private information. Most of the platforms allow for user to control which particular private information APIs the services can access and allow the end-user to accept or reject a particular privacy policy. However, the actual privacy practices of services may differ from the ones promised in the privacy policy. This paper outlines the basic assessment mechanisms and measures that enable user with insight on how much private information is actually being accessed and forwarded by each service deployed to the platform. The paper presents architecture for monitoring and assessing privacy practices of pervasive services deployed into a generic pervasive service platform. Furthermore, the paper presents platform specific privacy assessment implementation design for Java OSGi based pervasive platforms and describes an initial implementation and preliminary privacy assessment results, based on correlating data access events with data transmission events.**

Keywords-privacy; assessment; monitoring; pervasive; ubiquitous

I. INTRODUCTION

The ubiquity of smart phones and Internet connected devices with integrated sensing capabilities is resulting in more and more collecting, processing, aggregating and trading of personal information. As third party applications deployed on smart phones and internet connected devices can gain access to sensitive personal information, the amount of privacy threats and attacks is increasing (see [1], [2], and [3]).

The main focus of digital privacy protection is on a-priory protection minimizing and preventing the actual data release out of the data subject's realm. A lot of work has been done also on a-posterior privacy protection addressing protecting the data subject's interests and threats resulting from situations after the data have been released to the data controller [5]. Threats, misuse cases and generic a-posterior solutions for privacy in ubiquitous systems have been investigated in [4] and [6]. Lately, Hildebrandt et al. [7] have argued that in the new data intensive world it is not enough to merely stick to the minimization of data release and data concealment. The suggested solution is to keep all privacy practices and additionally increase transparency on how the data is collected and stored (see [7] for more information). As a result of this there has been an increasing trend in research to specify and prototype the so-called Transparency Enhancing Tools (TETs). Some recent research that is related to this paper includes the privacy logging tools that have been investigated by Hedbom et al. [8]. Furthermore, the system-wide dynamic taint tracking and analysis system capable of simultaneously tracking multiple sources of sensitive data has been investigated by the TaintDroid [9]. The system provides real-time analysis by leveraging Android's virtualized execution environment.

In this paper, we present a Privacy Transparency Enhancing ARchitecture (PEAR) along with initial implementation of Transparency Enhancing Tool (TET). The tool enables monitoring of privacy practices performed by third party services deployed on SOCIEITES ubiquitous platform [12] that is based on Virgo OSGi container. First, we describe the privacy monitoring and assessment architecture and describe how it relates to management of pervasive third party services deployed on a ubiquitous service platform. Then, we present initial set of privacy assessment mechanisms and privacy assessment visualizations.

II. PRIVACY MONITORING AND ASSESSMENT ARCHITECTURE

A typical service installed on user's personal mobile or other pervasive device can have access to a local database (Figure 1) where personal data is stored, including user related activity, location, and medical information. Mobile device's built-in sensors may continuously insert new data into the database. Other sensitive data such as credential storage and user profile can also be made available to the service. Services can then use the ability to communicate with local network or the internet and send any available data to other nodes and users.

To assess privacy practices of services, the privacy assessment component monitors any data access or data transmission by other services, calculates privacy assessment metrics, and outputs the assessment result to the end user (Figure 1). The user can then make an informed decision about any further actions against a service. The user can deny the service access to local data, disallow data transmission, or uninstall the service (Figure 1).

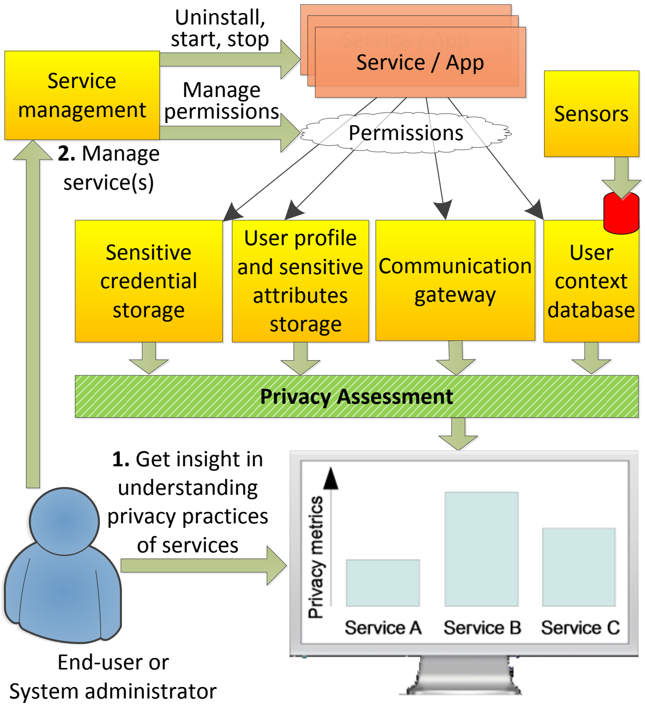


Figure 1. Interactions with the Environment

It should be noted that such a service is not necessarily a malicious service, or a service that violates the service level agreement (SLA). It is just a service that is suspected of accessing and/or transmitting user data more than it is necessary for its normal operation, whether the service behavior is in accordance with the SLA or not.

The approach shown in Figure 1 assumes a limited number of possibilities for a service to access or send data. These privacy monitoring points are integrated with privacy assessment component and whenever a service successfully uses these privacy monitoring points, the privacy assessment component is notified. From data properties, it can deduct and assess privacy practices of services.

Internally, privacy assessment component consists of four main building blocks shown in Figure 2. Arrows indicate data flow. The event logger is integrated with privacy monitoring points from Figure 1, collects events from the monitoring points and stores them into privacy practices log. The log itself is not available to other components and services because it contains sensitive information – the details about all recorded data access and transmission events.

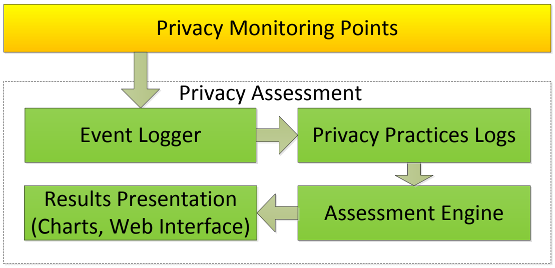

Privacy Monitoring Points
Privacy Assessment
Event Logger
Privacy Practices Logs
Results Presentation (Charts, Web Interface)
Assessment Engine

Figure 2. Internal Privacy Assessment Architecture

The assessment engine periodically parses the log, calculates privacy metrics for all software units or other entities in the log, and stores the results. The last block further aggregates results and exposes them to authorized external components, or displays them to the user directly.

A. Data Access Event

Any successful attempt to access user's data should be logged. Such data access events are not too hard to detect and log, assuming there are only a few possible ways to retrieve the data, and all of them are well-known. Examples include database access points and application-specific storage. Logging of data access events can be implemented by interception, such as with Aspect Oriented Programming (AOP) [13], or by integration of data access points and privacy assessment. The event details or features to store are at least those which may be available also for subsequent data transmission events and may help to correlate the events later. These details are typically time, data type, data size, and requestor identity (process ID, system user name, application-specific identity, etc.)

Collection of all that information alone can be problematic from the privacy point of view and the user needs to trust the privacy assessment software will not abuse the collected data.

B. Data Transmission Events

Capturing data transmission events is not as straightforward as capturing data access events. While there are only a limited number of data access points where particular data can be accessed, there are plenty possibilities (in terms of software) to transmit data over network.

However, there are cases where a single and well-known communication gateway is expected to be used. For example, in some pervasive environments the data receiver is not resolvable to an IP, email or other common address, and the communication gateway (or a limited number of communication gateways) offers a convenient way to transmit messages to specified receivers (Figure 7). While this is not a requirement for the proposed architecture, it enables feasible and efficient gathering of additional data, associated with a particular transmission event. Examples of such additional data include the environment-specific receiver endpoint, software module which transmitted the data, and transmitter identity if there are many possible identities with different authorization levels.

When it is required to detect and log all transmitted packets, it is necessary to use a traffic logger or parser on a lower level, which cannot be omitted when transmitting data. In such more general case where the logging point for data transmission is, e.g., on the TCP level (using a network traffic parser in Figure 7), less details about data transmission event can be gathered. This paper focuses on the former case, where a common gateway is assumed.

III. PRIVACY ASSESSMENT METRICS

The assessment results or privacy assessment metrics should give the user an estimated measure of how privacy

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-270-7

invasive each unit of software is. The software unit can be anything that can be mapped to a particular service. The given assessment values shall be directly comparable between multiple software units and also between multiple time intervals. Therefore, the metrics value should increase with number of data access and data transmission events. For a fixed number of data access and data transmission events, it should increase with estimated probability that the accessed data have in fact been transmitted. In addition to metrics, derived from correlations between the events, number of data access events and number of data transmission events themselves are important metrics. All these metrics are presented in details below.

Correlations are calculated from the available features from data access and data transmission events. First, correlations for individual event pairs are calculated for every available feature and labeled $c_{ij,k}$, where i is index of data access event, j is index of data transmission event, and k is feature index. An individual correlation c_{ij} is product of all feature correlations, e.g., time correlation and data size correlation for given events pair.

$$c_{ij} = \prod_{k=1}^{n} c_{ij,k} \qquad (1)$$

In Figure 3, all non-zero correlations between individual data access and transmission events c_{ij} are symbolized with dotted lines. Event pairs where data transmission precedes data access are obviously uncorrelated, and correlations c_{11}, c_{21}, c_{22}, c_{31}, c_{32}, c_{41}, c_{42}, and c_{43} are zero and not shown in Figure 3.

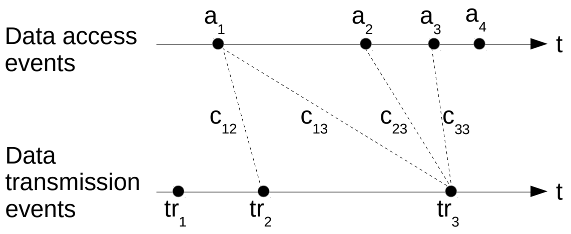


Figure 3. Events through time

Data size correlation function for individual event pairs is given by (2). The argument is difference in sizes of accessed data s_a and transmitted data s_t. If it was assumed that whenever a piece of data was accessed and then transmitted, its transmitted size was exactly the same as the size of original local data (ignoring even the addition of transmission protocol headers), then Kronecker delta function would be most appropriate. To amend for possible modification, aggregation, splitting, compression and encryption of the data before its transmission, the function is smooth, based on Gaussian curve, with asymptote greater than zero. Constants k_1 and k_2 present x-axis scaling and value at infinity, respectively. They can be adjusted to fine tune the algorithm.

$$c_{ij,1} = e^{-\left(\frac{s_t - s_a}{k_1}\right)^2} \cdot (1 - k_2) + k_2 \qquad (2)$$

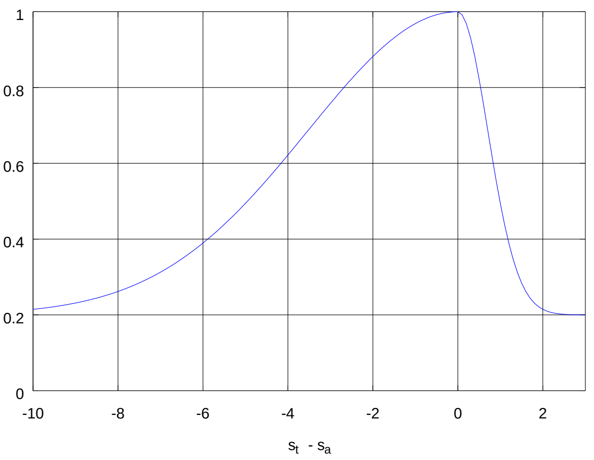


Figure 4. Correlation in data size for a single pair of events

Figure 4 shows an example of (2) where value of k_1 depends on sign of s_t - s_a. The example in Figure 4 is suitable for environments where data aggregation is less likely than data compression and splitting. For environments where services may often encrypt the data before transmission, it is even more important to adjust k_1 in this manner, because encryption usually includes compression.

Correlation in time is a completely different function, based on sigmoidal function of argument Δt or t_t - t_a, where t_t and t_a are times of data transmission and data access, respectively. Equation (3) presents the function for positive Δt values.

$$c_{ij,2} = \left(1 - \frac{1}{1+e^{-\frac{\Delta t}{k_3}-k_5}}\right) \cdot \frac{(1-k_4)}{\left(1-\frac{1}{1+e^{k_5}}\right)} + k_4 \qquad (3)$$

For negative argument values the correlation in time is zero (see Figure 5, for example) because a transmission event should not be correlated to any later data access event.

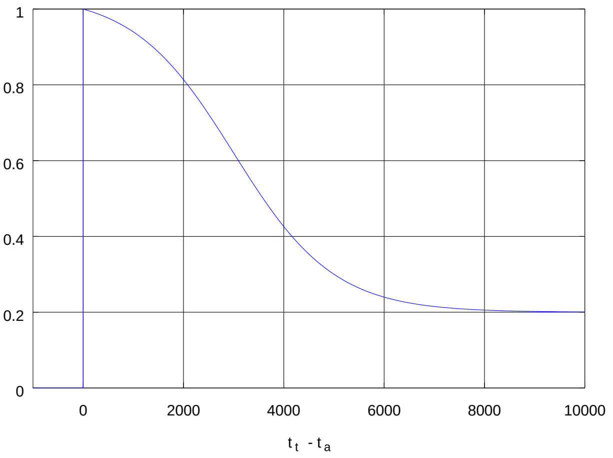


Figure 5. Correlation in time for a single pair of events

ISBN: 978-1-61208-270-7

The combined correlation for a single pair of events is calculated by (1). Additional features can be added where applicable. If only time and size features are available, then *n* from (1) equals 2. Correlations given by (1) are still intermediate results. Next step is to calculate correlation for a single data transmission event using (4). This is accomplished in two basic manners, which produce two or more results:

- Correlation with all data access events, i.e., for every *i* in (4).
- Correlation with only those data access events where the requestor matches the sender from the transmission event, i.e., for a limited set of *i* values. Depending on implementation, the requestor and sender can have one or more formats and meanings, leading to one or more distinct correlation results.

$$c_j = \sum_i c_{ij} \qquad (4)$$

The rationale for multiple correlations with data access events is availability of alternative routes from data retrieval to transmission (Figure 6). Correlation of a transmission event with only those data access events where the requestor matches data sender is the more reliable correlation in terms of least false positive errors. If it is known that component C_B transmitted data that has similar features to data D that had been retrieved previously by component C_A, then it is possible that component C_B transmitted data D. This possibility is more likely if it is known that C_A equals C_B, hence the greater reliability of such correlation. This correlation is most appropriate in cases where software components are assumed to retrieve the data and then transmit the data themselves. For events where this assumption is not satisfied (alternative route 2 in Figure 6), this correlation is more prone to false negative errors than correlation with all data access events.

The correlation with all data access events does not make that assumption, but is more likely to falsely correlate a data transmission event with data access events, i.e., it is more prone to false positive errors. This is especially problematic in a pervasive and sensor rich environment where multiple data are accessed and processed by third party services and may be independently transmitted to other parties. Such cases highlight the convenience, or even necessity of using a single communication gateway at high level where more features about the requestor and transmitter can be retrieved.

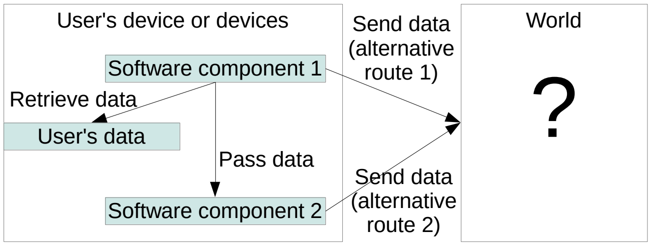


Figure 6. Alternative routes of data transmission

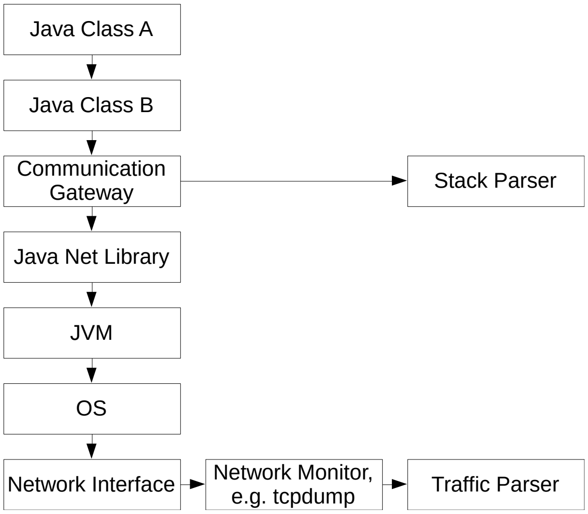


Figure 7. Data Transmission Detection

Regardless of the correlation type used in (4), final metrics for data sender *s* are calculated by summing those c_j where the sender associated with c_j matches *s* (5). This produces at least two different C_s values for each sender: at least one $C_{s,all}$, where (4) had been calculated for every *i*, and at least one $C_{s,matching}$, where (4) had been calculated for only those events where the requestor and sender match.

$$C_s = \sum_s c_j \qquad (5)$$

IV. Privacy Assessment implementation for Virgo

The Privacy Transparency Enhancing Tool has been developed for ubiquitous platform SOCIEITES [12], based on Virgo application server. In current implementation, the data access monitoring points and communication or sharing monitoring points have been realized.

The platform supports the concept of identities and each local service or a remote peer can be associated with an identity. Details such as requestor and sender identity and Java class name are captured for all data access and transmission events by parsing Java stack trace (Figure 7). On this platform it is very likely that services will use the common communication gateway for communication with other peers because it offers a convenient way to resolve remote identities and communicate with them. If this was not the case, a traffic parser on a lower level should have been used instead (Figure 7).

Java class name and identity values are stored as requestor in data access events and sender in data transmission events. This results in three distinct correlation types for a single data transmission event produced by (4):

- correlation with all data access events,
- correlation with only those data access events where the requestor class matches the sender class from the transmission event, and
- correlation with only those data access events where the requestor identity matches the sender identity from the transmission event.

These result in two final results for each sender class and two final results for each sender identity, which make four C_s result types from (5). The first two metrics are calculated for each class, and the last two for each identity:

- assessment metric for data sender class where all data access events were correlated,
- assessment metric for data sender class where only data access events with matching requestor class were correlated,
- assessment metric for data sender identity where all data access events were correlated, and
- assessment metric for data sender identity where only data access events with matching requestor identity were correlated.

For transmission events where data is passed to and transmitted by some other class (alternative route 2 in Figure 6), the value of correlation c_j from (4) with only data access events with matching class is zero. However, the correlation with events with matching identities amends for the missing information about passing the data between components. Correlation with those data access events where the requestor identity matches data sender identity implies a reasonable assumption – both components, services or classes that retrieve and transmit the data, do that under the same identity. In the unlikely situation where this is not the case, the metric derived from correlation with all data events still provides non-zero results.

V. Assessment Results

A. Presenting Results to User

Only high-level results calculated with (5) are shown to the user. A web based graphical user interface (GUI) shows results in form of bar charts, generated by the prototype (Figures 8-12). Figures 8, 9 and 12 show overall numbers of data access and data transmission events by identity and class name. Figure 9 shows the remote identities of data receivers. These basic results complement the metrics calculated by the assessment and help the user understand service behavior.

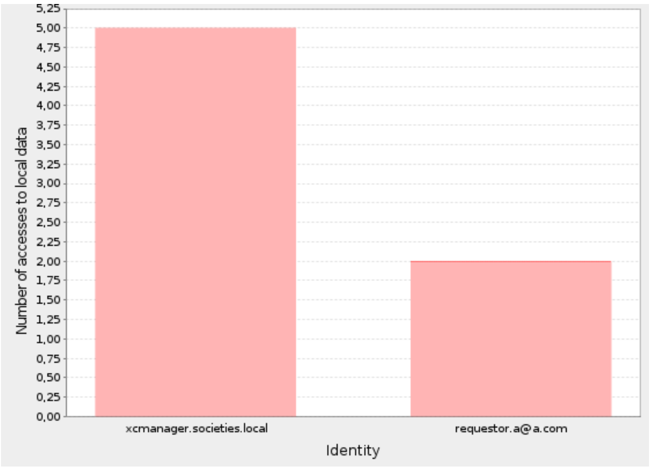


Figure 8. Local data access by identities

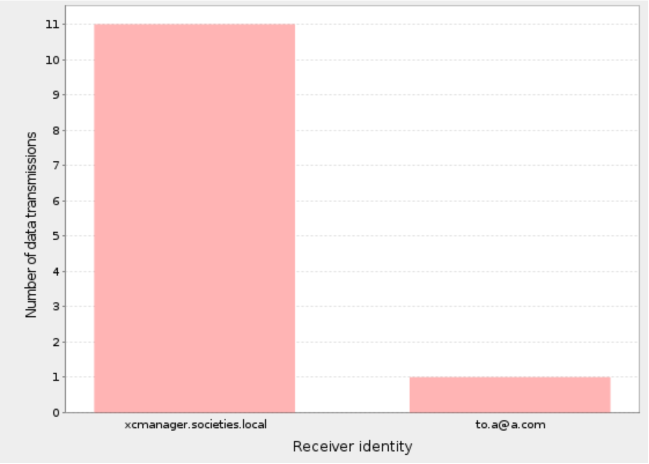


Figure 9. Data transmissions by receiver identity

Privacy assessment metrics for a particular class are shown in Figure 10, and Figure 11 shows the metrics for a particular identity. There are two classes that have transmitted data (the blue bars in Figure 10), but they have not accessed the data (value of both light-red bars is zero). However, this does not necessarily mean they have not transmitted local data because the non-zero blue bars indicate some data access events that had occurred before these classes transmitted data and the classes in Figure 10 might had received the data from other data requestor classes (route 2 in Figure 6).

Similar picture is shown in Figure 11, where the non-zero light-red bar shows the identity under which local data has been accessed at least once, and then at least once something has been transmitted under same identity. The higher blue bar for that identity shows there had been data access events by other identities, too. These data access events increased number of correlations by all data access events, i.e., the number of summands in (4).

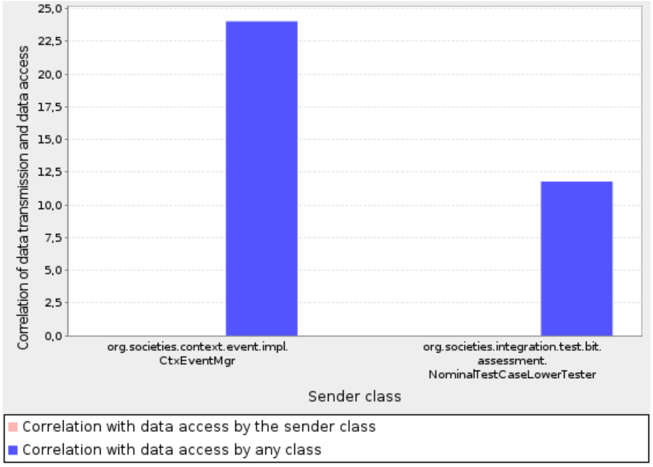


Figure 10. Privacy assessment results by Java class

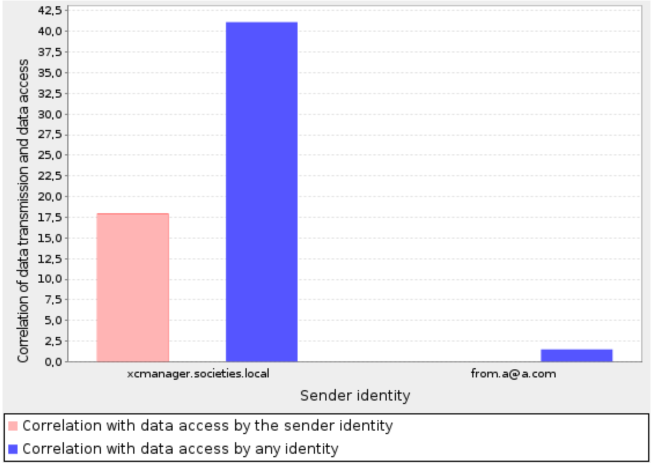


Figure 11. Privacy assessment results by identity

B. *Reacting to Results at Enforcement Points*

Seeing the privacy assessment results, the user can choose to react and impose limits on a particular service. To do that, enforcement points should be implemented at communication gateway, system firewall, and/or data access points (Figure 1). The more rigorous measure is to uninstall the services suspected of unnecessary user data transmission. Of course, there may always be services that access local data, receive and transmit data over network as part of their normal operation. A service with a low non-zero correlation may be even more suspicious than a service with higher correlation, e.g., if the former is not supposed to use network excessively or not at all.

VI. CONCLUSION AND FURTHER WORK

Privacy monitoring and assessment architecture was presented. The initial assessment analysis mechanisms based on correlating data access and transmission events have been described and the implementation of assessment visualization mechanisms has been presented. In the future we aim to extend the assessment method with more data type semantics and explore how current assessment mechanisms can support privacy risk and threat metrics combining our system with results from [10]. Additionally, we aim to port and adapt the prototype implementation for Android OS to support monitoring and assessing services deployed on smart phones.

ACKNOWLEDGMENT

Authors would like to thank the SOCIETIES project [11] consortium and EC for sponsoring the project.

REFERENCES

[1] L. Sheea, J.Alford, and R. Coffin, "Future of Privacy Forum Mobile Apps Study," http://www.futureofprivacy.org/wp-content/uploads/Mobile-Apps-Study-June-2012.pdf, June, 2012 [retrieved: March, 2013].

[2] M. Becher et al. "Mobile security catching up? Revealing the nuts and bolts of the security of mobile devices," Security and Privacy (SP), 2011 IEEE Symposium on. IEEE, 2011, pp. 96-111.

[3] A. Shabtai et al., "Google Android: A Comprehensive Security Assessment," IEEE Security & Privacy, March/April, 2010, pp. 35-44.

[4] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," Requirements Engineering, vol. 16, no. 1, 2011, pp. 3-32.

[5] C. Bryce et al., "Ubiquitous Privacy Protection: position paper," Proceedings of the 5th Workshop on Ubicomp Privacy in conjunction with the 9th International Conference on Ubiquitous Computing (UbiComp'2007), September, 2007, pp 397-402.

[6] K. Dolinar, J. Porekar, and A. Jerman Blazic, "Design Patterns for a Systemic Privacy Protection," The International Journal On Advances In Security, IARIA, 2009, vol2-3, pp. 267 – 287.

[7] M. Hildebrandt et al., "D 7.12: Behavioural Biometric Profiling and Transparency Enhancing Tools," FIDIS WP7 deliverable, http://www.fidis.net/fileadmin/fidis/deliverables/fidis-wp7-del7.12_behavioural-biometric_profiling_and_transparency_enhancing_tools.pdf [retrieved: March, 2013].

[8] H. Hedbom, T. Pulls, and M. Hansen, "Transparency Tools," Privacy and Identity Management for Life, doi: 10.1007/978-3-642-20317-6, Springer Berlin Heidelberg, 2011, pp. 135-143.

[9] W. Enck et al., "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," Proceedings of OSDI, 2010, pp. 393-407.

[10] R. Savola, "Towards a risk driven methodology for privacy Metrics development," Symposium on Privacy and Security Applications (PSA'10), August 2010, pp. 20-22.

[11] Self Orchestrating CommunIty ambiEnT IntelligEnce Spaces (SOCIETIES), EU FP7 project, Information and Communication Technologies, Grant Agreement Number 257493.

[12] M. Bordin, J. Floch, S. Rego, and A. Walsh, "D3.1: Service Model Architecture", SOCIETIES project WP3 deliverable.

[13] R. Johnson et al., "The Spring Framework – Reference Documentation," chapter 6, "Aspect Oriented Programming with Spring," http://static.springsource.org/spring/docs/2.0.x/reference/aop.html [retrieved: April, 2013].

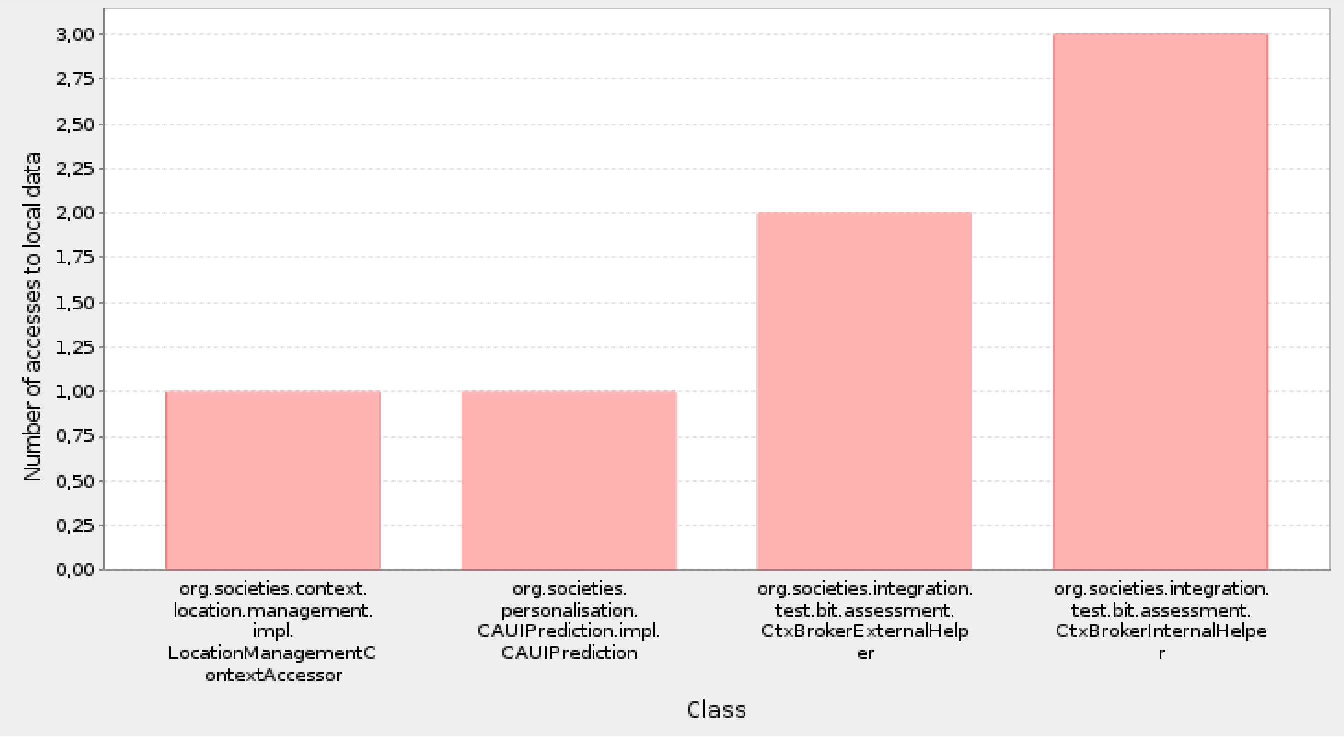


Figure 12. Local data access by Java classes

Towards Semantic-Supported SmartLife System Architectures for Big Data Services in the Cloud

Eman El-Sheikh
University of West Florida
Department of Computer Science
Pensacola, FL USA
eelsheikh@uwf.edu

Sikha Bagui
University of West Florida
Department of Computer Science
Pensacola, FL USA
bagui@uwf.edu

Donald G. Firesmith
Carnegie Mellon University
Software Engineering Institute
Pittsburgh, PA USA
dgf@sei.cmu.edu

Ilia Petrov
Reutlingen University
Data Management Lab
Reutlingen, Germany
ilia.petrov@reutlingen-university.de

Norman Wilde
University of West Florida
Department of Computer Science
Pensacola, FL USA
nwilde@uwf.edu

Alfred Zimmermann
Reutlingen University
Architecture Reference Lab
Reutlingen, Germany
alfred.zimmermann@reutlingen-university.de

Abstract **– SmartLife applications are emerging as intelligent user-centered systems that will shape future trends in technology and communication. The development of such applications integrates web services, cloud computing, and big data management, among other frameworks and methods. Our paper reports on new perspectives of services and cloud computing architectures for the challenging domain of SmartLife applications. In this research, we explore SmartLife applications in the context of semantic-supported systems architectures and big data in cloud settings. Using a SmartLife application scenario, we investigate graph data management, fast big data, and semantic support through ontological modeling. The ontological model and architecture reference model can be used to support semantic analysis and program comprehension of SmartLife applications.**

Keywords – SmartLife applications; Semantics and Ontology; Big Data Management; Enterprise Systems Architecture; Services–Oriented Architectures; Cloud Computing.

I. INTRODUCTION

Information and data are central components of our everyday activities. Social networks, smart portable devices, and intelligent cars, represent a few instances of a pervasive, information-driven vision we call SmartLife. Imagine speeding on the motorway, and receiving a text message on your mobile device from a friend asking to meet you. Since the number is on your personal contact list, the message is then transferred to your car's personal information system and read as well as displayed as soon as the traffic conditions allow that. You accept the invitation and the system checks recent social postings of your friend to recommend possible locations. Your friend has 'liked' multiple Espresso postings lately so the system infers that he/she might enjoy having one and executes a query for excellent Espresso places nearby and for times convenient for both your schedules. The SmartLife system would use your social profiles to recommend places. It would verify which of your close friends you have not met for a while are available and what they might have recommended, compile a simple list and display it on your car's head-up display. If you agree it will schedule an appointment in all personal calendars, distribute routes and queue them in the navigation systems, possibly recommending parking places. Your status messages will be automatically updated. Of course, you get to pick the best coffee blend and roast yourself!

The above is an example of a service-based semantically rich application scenario. Social graph analysis and management, big data, and cloud data management are essential to the above scenario. Ontological analysis, smart devices, personal information systems, hard non-functional requirements, such as location-independent response times and privacy, are some of the basic concepts in building such a SmartLife scenario.

Additional application domains of the SmartLife vision include: (i) intelligent mobility systems and services; (ii) intelligent energy support systems; (iii) smart personal health-care systems and services; (iv) intelligent transportation and logistics services; (v) smart environmental systems and services; (vi) intelligent systems and software engineering; (vii) intelligent engineering and manufacturing.

A. *Research Questions*

This paper describes work in progress that will address the following research questions:

1. How can service-oriented architectures (SOA) and enterprise systems architectures support SmartLife applications?
2. Is an ontological modeling approach useful to support semantic SmartLife applications?
3. How can semantic modeling approaches be effectively combined with system application

engineering, big data and services computing in the cloud?

4. How are semantic and social data in the SmartLife scenario efficiently managed as big data? How can real-time analysis and updates be efficiently performed on big graph data in cloud settings?
5. What are suitable approaches to enterprise IT-architectures for services and cloud computing, guiding the management and control of SmartLife scenarios?
6. How should existing software engineering methods evolve to cover SmartLife services?

B. *Impact*

The technological and business impact of the SmartLife vision has multiple aspects. While the business side targets intelligent approaches and structural business, the technological side is more diverse. Expected fields of innovation include:

- Software engineering methods for cloud applications
- Influence of dynamic configuration components for products, processes and systems
- Graph databases for fast big data and new hardware technologies
- Advanced architectural approaches for reconfigurable pervasive and mobile scenarios based on service-oriented and cloud computing architectures
- Common sematic approaches as a basis for modeling smart application scenarios for user-centered systems.

The rest of the paper describes the framework and methodology for the proposed research. Section II describes a minimalistic configuration scenario for SmartLife. Section III describes the implications and research issues resulting from SmartLife data management focusing on Big Graph Data Management. Section IV targets semantic representations and mechanisms for intelligent SmartLife support. Section V integrates both business and computer science aspects of a consistent configuration of enterprise systems architecture, and section VI concludes the paper.

II. SMARTLIFE APPLICATION SCENARIOS

SmartLife applications span a broad range of domains including intelligent configuration services, intelligent transportation and logistics services, personal health care systems and services, smart environmental systems, and intelligent engineering and manufacturing systems. Below we describe a simple starting scenario for SmartLife.

WebAutoParts is a hypothetical online automobile parts dealer intended to model an Internet start-up company that is using SOA for rapid development [1]. Its software uses BPEL for orchestration of commercially available external services from well-known vendors. As shown in Figure 1, the Order Processing workflow for WebAutoParts has two stubbed in-house BPEL services (OrderProcessing and InventoryRepository) and four commercially-available external services: Amazon Web Services - *SimpleDB* (data base) and *SimpleQueueService* (message queuing); StrikeIron.com - *TaxDataBasic* (sales tax rates); Ecocoma - *USPS* (shipping costs).

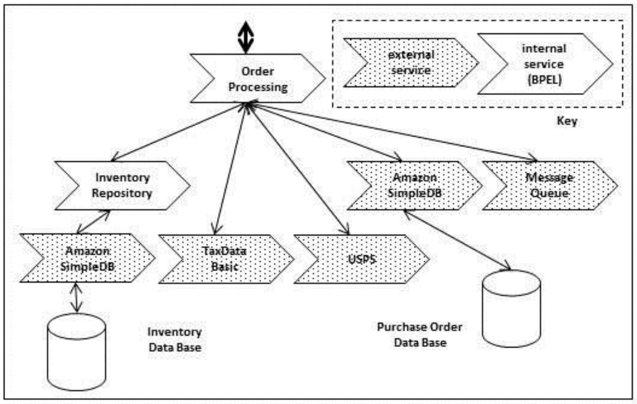


Figure 1. WebAutoParts: Services in the Order Processing Workflow

WebAutoParts is much smaller than most real SOA applications. However, it is useful for ontological exploration since it consists of syntactically correct BPEL code and contains XSD and WSDL documents typical of current industrial practice. We will also explore the use of other SOA systems for SmartLife domains, such as intelligent transportation services and intelligent engineering and manufacturing systems in future research work.

III. GRAPH DATA MANAGEMENT

In terms of data management, SmartLife systems manage and analyze big data. Major components are social, enterprise, semantic, and sensor data [2]. We actively investigate the following research aspects:

- Heterogeneity. Data from multiple possibly heterogeneous data sources have to be federated. New approaches to data fusion and cleaning are needed.
- High Volume. Data is being produced at high rates on the scale of Petabytes or Exabytes from many users and data sources. In contrast to traditional data management it is not reasonable to assume upper bounds. It is mostly distributed.
- High Update Rates. Data and content are being produced at high rates for new activity (tweets, social graph updates and social content, sensor and mobile data). Hence a paradigm shift is required in big data management and high update rates.
- Near/Real-time Analytics. Near-time analytics and discovery are a prerequisite for successful SmartLife systems. In addition to traditional analytics, data mining and information retrieval, SmartLife systems are expected to offer user recommendations. The near-time character of data-analysis requires new approaches.

A research area that gains significant attention and offers a common way of data processing and analysis is graph data management. Existing approaches, algorithms and systems need to be reevaluated in the above context [3]. The use of novel hardware such as many-core CPUs, FPGAs, new storage technologies, like Non-Violate and Flash memories, are critical to handle the high update rates and near-time analytics.

One research goal is to investigate graph database systems [4] in a cloud setting that handle huge data volumes and high update rates and at the same time offer near-time analytics, recommender functionality and crowdsourcing. The efficient use of new hardware technologies is another key research goal.

Social big data imply processing of very large graphs that are typically maintained at multiple sites (cloud settings) with high update rates [5]. Traditionally high volume graph data is being handled by disc-based graph databases, which are too slow to handle the complexity of the typical inference and analytics graph queries. Low response times represent a key non-functional requirement. Additional performance related research issues arise from the need to handle mixed loads – complex graph analytics as well as high update rates. The efficient use of new hardware is a key requirement to meet these performance challenges, which translates into a number of research issues: (i) optimal use of flash and non-volatile memories since many of the current algorithms are not suitable; (ii) efficient use of multi-core CPUs and FPGAs for graph data analysis; and (iii) distribution and synchronization problems in Cloud settings.

IV. Semantic Support Through Ontologies

A. *Development of an Ontological Model*

The Open Group developed and released the SOA Ontology 2.0 [6]. This ontology has two main purposes:

1. It defines the concepts, terminology and semantics of SOA in both business and technical terms.
2. It contributes to model-driven SOA implementation.

The Open Group’s SOA Ontology [6] is represented in the Web Ontology Language. The Open Group ontology contains classes and properties corresponding to the core concepts of SOA. The formal OWL definitions are represented (i) in the OWL syntax; (ii) as UML models of the concepts and their relationships; and (iii) all models are supplemented by natural language descriptions.

B. *Ontological Model for SmartLife Applications*

The phase after development of the ontological model will focus on exploring how the model can be used to support program comprehension for SOA-based SmartLife systems. Several specific SOA comprehension tasks will be identified, including (but not limited to):

- Impact analysis (If X is changed, what additional changes may be needed?)
- Concept location (Where is concept Y implemented in this system?)

We will explore visualization of the ontological model developed to support: (i) system comprehension and (ii) information and data management. Additional research questions include: Is the ontological reference model sufficient to model a SOA SmartLife system? Are there gaps? Would the reference model need to be extended?

C. *Comparing the Ontological Approach to Other Knowledge Modeling Approaches*

1) Concept Maps

Concept maps are an established framework for organizing and representing knowledge [7]. A concept map is a diagram that shows the relationships among concepts. Concepts, usually represented as boxes or circles, are connected with labeled arrows in a downward-branching hierarchical structure. The relationship between concepts can be expressed in linking phrases such as "is" or "includes." Concept maps are particularly useful for analyzing and organizing large and complex domains. Concept maps can be structured hierarchically, linked together, and augmented with other resources such as text, graphics, videos, etc., to create a knowledge model [8].

2) Entity Relationship Model

The Entity Relationship (ER) model is an established data modeling technique, well accepted in the database world [9]. The ER model is used to visually represent data in databases in terms of entities, their attributes and relationships. Entities describe a complex structured concept like a person, place, thing or event of interest. Attributes are used to describe entities. Attributes can be either single value or multi-valued. And, relationships describe associations among entities. Relationships are explained in terms of their connectivity (or cardinality), and their connectivity can be indicated by one-to-one (1:1), one-to-many (1:M) and many-to-many (M:N) relationships. Cardinality is related to upper and lower bounds. Participation in this connectivity by member entities may be optional (partial) or mandatory (full).

3) Unified Modeling Language Model

Unified Modeling Language (UML) is a standardized general-purpose modeling language used in object-oriented software engineering. The standard is managed, and was created, by the Object Management Group.

UML is used to specify and visualize the artifacts of an object-oriented software-intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as: activities, actors, business processes, database schemas, and (logical) components, programming language statements, reusable software components. UML combines techniques from data modeling (ER modeling), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.

4) Tree Abstractions

The tree representation is a hierarchical representation of the data, mainly used in the XML data format [10]. This structure allows representing information using parent/child relationships: each parent can have many children, but each child has only one parent (also known as a 1-to-many relationship). All attributes of a specific record are listed under an entity type.

We are exploring the use of concept maps and other knowledge models for semantic analysis of SmartLife applications. The ontological model developed for the target applications can be compared to a knowledge model of the application to identify similarities and differences between the two program comprehension approaches, as well as strengths and weaknesses of each approach.

V. Enterprise Systems Architecture

In areas where flexibility or agility in business is important, services computing is the approach of choice to organize and utilize distributed capabilities. Innovation oriented companies have introduced in recent years service-oriented architectures to assist in closing the business - IT gap and making it cloud-ready. The benefits of SOA are recognized for systems on the way to cloud computing and being ready for extended service models. They comprise flexibility, process orientation, time-to-market, and innovation.

A. Reference Architectures for Services & Cloud Computing

The OASIS Reference Model for Service Oriented Architecture [11] is an abstract framework, which guides reference architectures [12]. The ESARC – Enterprise Services Architecture Reference Cube [13] (Figure 2) is more specific and completes these architectural standards in the context of EAM – Enterprise Architecture Management, and extends these architecture standards for services and cloud computing.

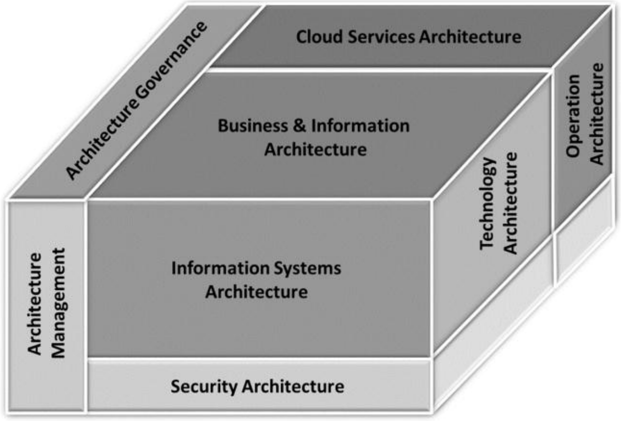


Figure 2. ESARC - Enterprise Software Architecture Reference Cube

ESARC provides an abstract model for application architectures and implementation of service-based enterprise systems. ESARC is an original architecture reference model, which provides an integral view for main interweaved architecture types. ESARC abstracts from a concrete business scenario or technologies. The Open Group Architecture Framework provides the basic blueprint and structure for our extended service-oriented enterprise software architecture domains like: Architecture Governance, Architecture Management, Business and Information Architecture, Information Systems Architecture, Technology Architecture, Operation Architecture, and Cloud Services Architecture. ESARC provides a coherent aid for examination, comparison, classification, quality evaluation and optimization of architectures.

The Business and Information Reference Architecture - BIRA (Figure 2) provides, for instance, a single source and comprehensive repository of knowledge from which concrete corporate initiatives will evolve and link. This knowledge is model-based and defines an integrated enterprise business model, which includes organization models and business processes. The BIRA opens a connection to IT infrastructures, IT systems, and software as well as security architectures. The BIRA confers the basis for business-IT alignment and therefore models the business and information strategy, the organization, and main business demands as well as requirements for information systems, such as key business processes, business rules, business products, services, and related business control information.

The ESARC Information Systems Reference Architecture –ISRA (Figure 2) is the application reference architecture and contains the main application-specific service types, defining their relationship by a layer model of building services. The core functionality of domain services is linked with the application interaction capabilities and with the business processes of the customer organization. In our research we are integrating the reference models for services computing [13].

Cloud architectures are still in development and have not yet reached their full potential of integrating EAM with Services Computing and Cloud Computing. Integrating and exploring these three architectural dimensions into consistent reference architectures is a basic part of our current research. The ESARC – Cloud Services Architecture (Figure 2) provides a reference-model-based synthesis of current standards and reference architectures, like [14].

B. Architecture Metamodel and Ontology

Metamodels are used to define architecture model elements and their relationships within ESARC. We use metamodels as an abstraction for architectural elements and relate them to architecture ontologies [15]. The OASIS Reference Model for SOA [11] is an abstract framework, which defines generic elements and their relationships for service-oriented architectures. This reference model is not a standard, but provides a common semantic model for different specialized implementations.

Reference architectures [12] are derived from a reference model. It is a composition of related architectural elements, which are built from typed building blocks as the result of a pattern-based mapping of reference models to software elements. Architecture patterns, as in [17], [18] are human readable abstractions for known architecture quality attributes, and represent standardized solutions, considering architectural constraints for certain recurring problems.

Architecture ontologies represent a common vocabulary for enterprise architects who need to share their information based on explicitly defined concepts. Ontologies include the ability to automatically infer transitive knowledge. The technical standard of service-oriented architecture ontology from [6] defines core concepts, terminology, and semantics of a service-oriented architecture in order to improve the alignment between the business and IT communities. The following stakeholders are potential users of the SOA ontology, related architecture metamodels, as well as concrete architectural building blocks: business people and business architects, information systems and software architects, architects for the technological infrastructure, cloud services architects and security architects. The metamodel of BIRA consists of ESARC-specific concepts, which are derived as specializations from generic concepts such as Element and Composition from the Open Group's SOA Ontology [6].

Using the ESARC ontology, we can navigate in the multidimensional space of enterprise architecture management structures and enable in a future research effort of semantic-supported navigation for architects as well as intelligent inferences. Additionally we want to add visualizations for these ontology concepts, as part of a sematic-supported architecture management cockpit.

C. *Methodology Framework for System Architectures*

As its name implies, the Method Framework for Engineering System Architectures (MFESA) [16] is a framework for using situational method engineering to create appropriate methods for engineering system architectures. MFESA consists of:

- An ontology that defines the concepts underlying system architecture engineering
- A metamodel that defines the foundation classes of the method components
- A repository of reusable method components derived from the foundation classes of the metamodel
- A metamethod for constructing system architecture engineering methods by selecting, tailoring, and integrating method components from the MFESA repository.

The Quality Assessment of System Architectures and their Requirements (QUASAR) is a method for assessing the quality of system architectures and architecturally-significant quality requirements. QUASAR is based on the concept of requirements- and architecture-level quality cases consisting of:

- Claims – developers' assertions that the (a) architecturally-significant quality requirements are sufficiently complete, correct, consistent, etc. and (b) architecture is sufficiently complete and meets the architecturally-significant requirements
- Arguments – clear, compelling, and relevant developer arguments that sufficiently justify the assessor's belief in the developers' *claims* (e.g., architectural decisions, inventions, engineering trade-offs, assumptions, and associated rationales)
- Evidence – adequate, credible, and official substantiation supporting the developers' *arguments* (e.g., architectural diagrams, models, and documents).

D. *Patterns and Repository for Architecture Diagnostics and Optimization*

Our pattern language for architecture assessments of service-oriented enterprise systems [17] provides a procedural method framework for architecture assessment processes and questionnaire design. We organize and represent our architecture assessment patterns according to the structures of the architecture maturity framework SOAMMI [13], [18]: *Architecture Domains*, *Architecture Areas*, *Problem Descriptions* - associated with *Specific Goals*, *Solution Elements* that are connected to *Specific Practices* and *Related Patterns*, which are subsequent connections of applicable patterns within the pattern language.

Linking elements to specific practices of the SOAMMI framework indicate solutions for architecture assessments and improvements of service-oriented enterprise systems. This assessment and improvement knowledge is both verification and design knowledge, which is a procedural knowledge based on standards, best practices, and assessment experience for architecture assessments of service-oriented enterprise systems. It is therefore both concrete and specific for setting the status of service-oriented enterprise architectures, and helps to establish an improvement path for change.

We have identified and distinguished a set of 43 patterns as parts of a newly designed pattern language in the context of 7 Architecture Domains and 22 Architecture Areas. Even though our architecture quality patterns accord to the Specific as well as the Generic Goals and Practices of the SOAMMI framework, they extend these structures by navigable patterns [18], as part of an architecture assessment language. This pattern structure enables architecture quality assessors to navigate bi-directionally, to support both diagnostics and optimization processes, as well as to provide a clear link to questionnaires.

E. *Enterprise Architecture Governanace and Management*

Architecture Governance defines and maintains the Architecture Governance cycle [13]. It sets the abstract governance frame for concrete architecture activities within the enterprise or a product line development and specifies the following management activities: plan, define, enable, measure, and control. The second aim of Architecture Governance is to set rules for architecture compliance to internal and external standards. Enterprise and software architects are acting on a sophisticated connection path emanating from business and IT strategy to the architecture landscape realization for interrelated business domains, applications and technologies. Architecture Governance has to set rules for the empowerment of people, defining the structures and procedures of an Architecture Governance Board, and setting rules for communication. We specify architecture governance models for concepts such as: 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.

VI. CONCLUSION

SmartLife applications are emerging as intelligent user-centered systems that will shape future trends in technology and communication. The development of such applications integrates web services, cloud computing, and big data management, among other frameworks and methods. The basic approaches within each field are already well known and used. However, such methods are not directly applicable and properly integrated for SmartLife applications. Existing approaches can be extended to exploit synergistic effects resulting from the SmartLife context. Technological evolution is also expected forming a feedback cycle from SmartLife scenarios to new technologies.

We have set up a transatlantic, multi-institutional research cooperation starting with this project, which would be extended to related areas as well as to student and academic exchanges and common publication efforts in conferences and journals. This paper described the framework and methodology for the research in progress. We explore SmartLife applications in the context of semantic-supported systems architectures and big data in cloud settings. Using a SmartLife application scenario, we investigate graph data management, fast big data, and semantic support through ontological modeling.

We have developed a prototype SmartLife application, WebAutoParts, to use as a test bed for our research project. We are exploring how the semantic and social data in the SmartLife scenario can be efficiently managed as big data, and how real-time analysis and updates can be efficiently performed on big graph data in cloud settings.

In addition, we have defined the ontological and architectural reference frameworks for our target SmartLife application, and are currently working on developing the ontological model for this application. Future work includes analyzing how the ontological and architecture models developed can be used to support semantic analysis and program comprehension of SmartLife applications. The models can be compared to and combined with other semantic modeling approaches to support development and maintenance of SmartLife applications.

ACKNOWLEDGMENTS

Work described in this paper was partially supported by the University of West Florida Foundation under the Nystul Eminent Scholar Endowment, and the SOA Innovation Lab Germany.

REFERENCES

[1] T. Reichherzer, E. El-Sheikh, N. Wilde, L. White, J. Coffey, and S. Simmons, "Towards intelligent search support for web services evolution: identifying the right abstractions", Proceedings of 2011 13th IEEE International Symposium on Web Systems Evolution (WSE), 30 Sept. 2011, pp. 53-58.

[2] J. Gray, A. Szalay, "Science In An Exponential World". Nature, , 23 March 2006, V. 440.23.

[3] Cheng, J., Ke, Y., and Ng, W.: Efficient query processing on graph databases. ACM Trans. Database Syst. 34, 1, Article 2, April 2009.

[4] Frischbier, S., Petrov, I.: Aspects of Data-Intensive Cloud Computing. From Active Data Management to Event-Based Systems and More, 2010, pp. 57-77.

[5] R. Sears, R. Ramakrishnan, "bLSM: a general purpose log structured merge tree", In Proc. of SIGMOD 2012.

[6] Open Group, "Service-Oriented Architecture Ontology", Technical Standard, The Open Group, 2010.

[7] J. Novak, and D. Gowin, "Learning How to Learn", Cambridge. University Press, New York, NY, 1984.

[8] J. W. Coffey and T. Eskridge, "Case Studies of Knowledge Modeling for Knowledge Preservation and Sharing in the U.S. Nuclear Power Industry", Journal of Information and Knowledge Management. 7(3), 2008, pp. 173-18.

[9] S. Bagui and R. Earp, (2012). "Database Design Using ER Diagrams", 2nd edition, Taylor and Francis, 2012.

[10] S. Bagui, "Mapping XML Schema to Entity Relationship and Extended Entity Relationship Models", International Journal of Intelligent Information and Database Systems, 3(4), 2007, pp. 325-345.

[11] 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.

[12] Open Group "SOA Reference Architecture", The Open Group, 2011.

[13] 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, pp. 592-599.

[14] M. Behrendt, B. Glaser, P. Kopp, R. Diekmann, G. Breiter, S. Pappe, H. Kreger, and A. Arsanjani, "Introduction and Architecture Overview – IBM Cloud Computing Reference Architecture 2.0", IBM, 2011.

[15] A. Zimmermann, and G. Zimmermann, "Enterprise Architecture Ontology for Services Computing", SERVICE COMPUTATION 2012: Nice – France – July 22-27, 2012, ISBN 978-1-61208-215-8, pp. 64-69.

[16] D. G. Firesmith with P. Capell, D. Falkenthal, C. B. Hammons, D. Latimer, and T. Merendino, "The Method Framework for Engineering System Architectures", CRC Presstaylor & Francis Group, 2009.

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

[18] A. Zimmermann, F. Laux, and R. Reiners, "A Pattern Language for Architecture Assessments of Service-oriented Enterprise Systems", PATTERNS 2012: Nice – France – July 22-27, 2012, ISBN 978-1-61208-158-8, 2011, pp. 7-12.

A New Process Model for Optimizing IT Outsourcing Operations in the German Automotive Industry

Christine Brautsch
Department of Car IT
AUDI AG
Ingolstadt, Germany
christine.brautsch@audi.de

Martin Wynn
School of Computing and Technology,
University of Gloucestershire,
Cheltenham, UK
mwynn@glos.ac.uk

***Abstract* - The outsourcing of IT services is a significant business activity for many companies and is a well-established element of services management worldwide. However, the process is neither well defined nor understood in many industries, including the automotive sector, where it is of growing importance. A review of existing literature reveals consideration of specific aspects of outsourcing in isolation, but relatively little material that provides a comprehensive framework for analysis. This paper thus identifies the main stages in IT outsourcing operations in the German automotive industry and seeks to establish the critical success factors that can help ensure quality outcomes. It suggests a clear definition of IT outsourcing and constructs a new conceptual process model, that provides the basis for a range of analytical materials to complement the existing literature, and which will also be of value to practitioners working in this field.**

Keywords - IT outsourcing; operations; success factor; service provider; process mode; CSFs

I. INTRODUCTION

Outsourcing as a business process or function is not new, especially in the field of information technology, systems and services, in which it has become increasingly important during the last few years [1]. In this context, when referring to 'IT (information technology) outsourcing', we assume the broader definition of IT that encompasses technologies, systems and services. "Information technology, in its narrow definition, refers to the technological side of an information system. It includes hardware, databases, software networks, and other devices. ... sometimes, the term IT is also used interchangeably with IS (information systems), or it may even be used as a broader concept that describes a collection of several information systems, users, and management for an entire organisation" [2].

The reasons for outsourcing are various, but potential cost savings and the transparency of costs are often to the fore [3]. The business case for IT outsourcing can contain a range of possible benefits, including economies of scale in the use of, for example, third party hardware and infrastructure, and the freeing up of 'human capital' to concentrate on core business activities or strategic IT issues. Nevertheless, many IT outsourcing operations fail or require renegotiation within the project life-cycle [4]. Groetschel [5] suggests that 30 % of all IT outsourcing projects fail or the expected results are not reached. Furthermore, it seems that IT outsourcing success is generally the expected outcome and the risks are often underestimated.

The research centres on IT outsourcing in the German automotive industry, where the process of IT outsourcing is neither well documented nor researched. It is a dynamic technology environment with the life cycles for both software and hardware products continually shortening because of new developments [6]. In addition, the development of new technologies relating specifically to automobiles requires the integration of IT professionals and auto technology specialists. This has resulted in a rapid evolution of the IT function within the automobile industry to encompass technologies that are outside its traditional limits [7]. Technological and strategic decisions are taken within the organization, and a web of suppliers produces the components, which are often integrated within the car. The need for alignment of technology and business strategies is paramount.

Inevitably, these developments influence sourcing strategy. The engagement of specialized companies as business partners may be viewed positively as these firms have developed efficient structures for planning and controlling the entire IT life-cycle [1] [5]. Furthermore an "effective knowledge-sharing process over organizational boundaries" can ensure the right flow of information between outsourcing partners [8].

Within this dynamic technological and business context, this paper addresses three research questions (RQs):

RQ1. What are the main stages in the IT outsourcing process in the German automotive industry?

RQ2. What are the critical success factors (CSFs) for IT outsourcing performance in the German automotive industry?

RQ3. Can a new process model be constructed for IT outsourcing in the German automotive industry that helps ensure quality outcomes and deliverables?

The paper first reviews existing literature to establish a theoretical framework for the initial research activity (section II). Research philosophy and methodology are then considered (section III), before the research questions are fully addressed, focusing particularly on a new process model for IT outsourcing in the German automotive industry (section IV). The concluding section (section V) puts forward a new definition for IT outsourcing and suggests a hierarchy of success factors that impact upon the outsourcing process, looking also at future research activity in this field.

II. THEORETICAL FRAMEWORK

The analysis of the existing literature shows a paradigm shift in practice in the automobile sector as regards outsourcing: companies no longer ask themselves “Shall I outsource parts of my IT?” the question today is rather “How can I realize IT outsourcing in the best way for my company?” [9]. This has led to a more individual and more focused analysis of IT outsourcing options as measured against future business challenges.

The literature review suggests that the persons responsible for IT outsourcing often underestimate the complexity of the entire process. There may be a concentration on particular aspects (e.g. based on their professional background) without an appreciation of the whole process [10]. An important aspect in this context seems to be the need for clear objective formulation within each IT outsourcing project.

The review revealed two main trends in the automotive sector: first, a general increase in IT outsourcing as companies strive for improved cost-effectiveness; second, an increasing number of IT outsourcing projects fail or the expected results are not reached. The corollary to this second point is that many companies assume IT outsourcing will be successful and risks are often underestimated.

The entire IT outsourcing life-cycle is not widely analysed in the academic literature but existing studies allowed the development of a provisional life cycle model or conceptual framework that has been tested out and refined through primary research. A conceptual framework can be seen as a type of intermediate theory that attempts to connect all aspects of inquiry, i.e. problem definition and literature review, as well as methodology development, data collection and interpretation. In general, conceptual framework development has often been linked to exploratory types of research [22]. The constructed framework consists of eight key-stages and has been visualized in the form of a life-cycle (based on the plan, build and run approach – Fig. 2).

Analysis of the existing literature also allowed an identification of possible critical success factors (CSFs) that could be provisionally allocated to different stages in the life-cycle (Table I). In determining these factors, the different opportunities and risks within IT outsourcing were identified from the existing literature as well. These success factors suggested the potential measures that might be used to assess the degree of achievement in outsourcing projects.

Table I. Possible Critical Success Factors identified from existing literature

CSF1	Transparency concerning: - strategic orientation, - internal IT performance, - customer requirements	[11, 12, 13, 14, 15]
CSF2	Economic point of view: value-for-money (management of costs)	[11, 13, 15, 16]
CSF3	Transparency within the selection process of a provider (choosing the right service provider)	[11, 13, 17]
CSF4	Proper communication and information	[11, 13, 15, 18]
CSF5	Contract management and controlling	[11, 13, 15, 17]
CSF6	Service control and lessons learned (e.g. define service level agreements)	[11, 13, 15, 20, 21]

The literature review thus helped develop, but also provided some tentative answers to, the three main research questions for the primary research phase of the project. In general, the concept of IT outsourcing has been explored in a way that will generate a contribution to knowledge, i.e. advancing the understanding of the concept, and/or improving the outsourcing processes. This will be the basis for the development of a new process model for IT outsourcing in the automotive industry in Germany.

III. RESEARCH PHILOSOPHY AND METHODOLOGY

The research is based on linking practice and theory in a pragmatic way and is guided by practical experiences [23], [24], [25]. Indeed, the research will be based on the paradigm of pragmatism. “What?” and How?” become the main aspects of the research problem [26]. The research follows a step by step analysis for each RQ [27]. The research is a single in depth case-study based on qualitative research, which will be used to develop and justify a conceptual framework [28]. The information is being obtained through semi-structured interviews, which provide the opportunity to speak directly to stakeholders. Furthermore they allow different views to be examined very closely and in depth.

The analytical techniques for qualitative data are not well developed, as this kind of data consists of interpretable words and observations, and not numbers [28]. According to Taylor and Renner [29], qualitative data analysis (of narrative data) requires creativity, discipline and a systematic approach. Thus there is no single solution - the results depend on the RQs and available resources [29].

Based on eight different stakeholder-groups (Fig. 1), interviews were held in a two-step approach reflecting the RQs. The first stage enquired about the different stages of outsourcing projects to build a basis for further discussions. At the end of each interview, the findings from the literature were presented and discussed with the participants. In

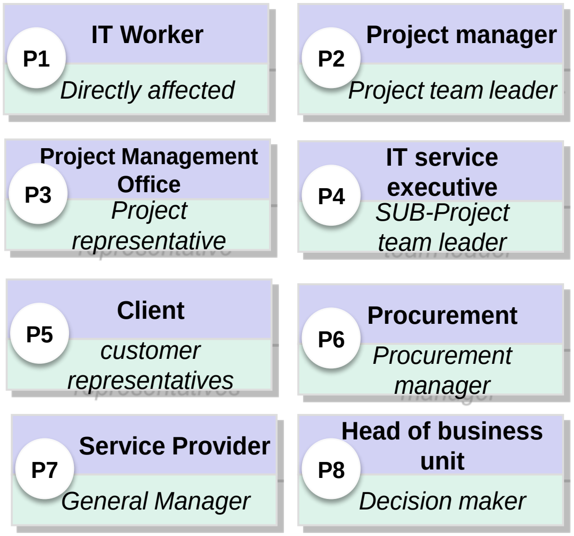


Figure 1. The Eight Stakeholder Profiles (P1-P8)

addition, each interview was completed with a self-allocation of the participants´ business activity in the outsourcing life-cycle. This ensured that all areas of the life-cycle were covered and queried more than once. Deviations from the set agenda can be made in order to explore new and particularly interesting points raised in the course of each interview [32].

The second stage concentrates on the CSFs for outsourcing delivery. The interviews lasted about two hours each, one hour per stage. The first person interviewed was the IT worker to get a general overview and to validate the pertinence of the questions [32]. The last interview was with the most experienced person (head of the business department) to elucidate uncertainties (until data saturation is reached). Finally, in order to verify the results, the developed model was sent to all participants for comments and will be sent to the company for final approval [32]. The analysis of the gathered data is being fully incorporated in a refined conceptual framework. In summary, the systematic literature review indicated potential critical success factors; the interviews gave the chance to validate and test these initial findings in order to come up with a solid set of factors for success in IT outsourcing.

IV. Towards a New Process Model for IT Outsourcing in the Automotive Industry

The primary research phase of this project focuses on eight interviews that have been undertaken as sub-cases. All interviewees have been anonymised, using the acronym P1 – P8, but Fig. 1 provides job titles and functions in the overall process. Analysis of this interview material suggests the following answers to the research questions.

A. *Research Question 1: What are the main stages in the IT outsourcing process in the German automotive industry?*

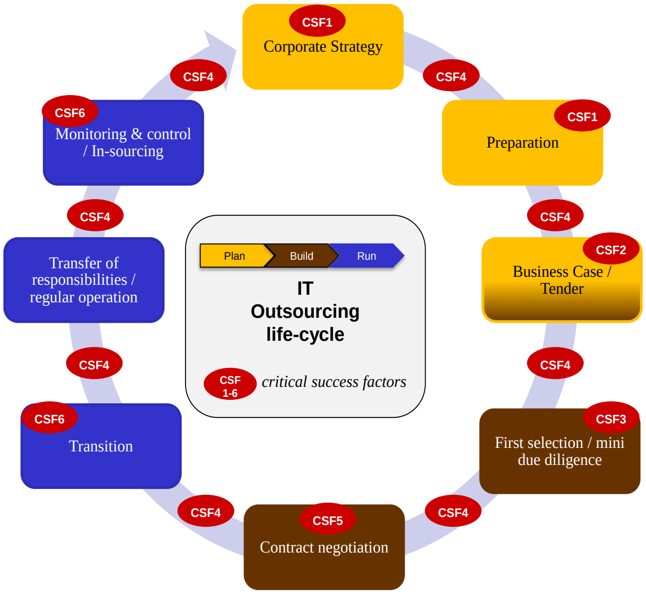


Figure 2. Initial Conceptual Model based on Plan-Build-Run Concepts

Interviewees were asked for their personal understanding of IT outsourcing. In general, the poor definition suggested in the literature review was confirmed. Each participant gave a different definition of the term “IT outsourcing”. However, there was some common ground in all interviewees’ perceptions, notably that the concept involves a transfer of a defined set of tasks to an external service provider. The main stages of the life-cycle garnered from the literature review were generally confirmed as logical and realistic. There is a correlation between the field of activity of the interviewee and his understanding of the overall process. Based on his practical experience and field of activity, P1, for example, described IT outsourcing as starting after the end of the ‘build-stage’. For P1, IT outsourcing is a corporate strategic objective, which *has* to be realized. Consequently, P1 has no influence on the outsourcing decision itself. However, P1 is charged with determining ‘how’ the new service will be delivered. This implies that P1 has, in the main, two dimensions of activity:

- Design of the operational service (e.g. meeting structure, communication channels, setting up an operational manual)
- Setting up and measurement of service level agreements (SLAs) with service providers.

P4, however, has a very content-driven understanding of the life-cycle. For P4, IT outsourcing consists of twelve stages, starting with the analysis of the service providers’ offers and ending with the completion of the transition to the third party and associated knowledge transfer process. The influence of professional background can again be seen in P4’s perception of the overall process. Somewhat to the contrary, three interviewees (P2, P3, and P8) had a more top level or superficial view of the process.

Overall, however, the conceptual framework and the provisional stages in the outsourcing process were generally confirmed in the first hand interviews, with the addition of three new concepts or developments, and one new stage.

The first one new concept concerns the influence of mental models or preconceptions that each stakeholder may have regarding the outsourcing process. For example, P3 stated that “…the specific understanding of circumstances that people have, has to be considered continuously during the entire process starting even before the strategy is agreed…”

Second, some interviewees suggested that the transition stage could usefully be subdivided into two stages: transition and transformation, and this is reflected in the new model with transformation being a new stage 7 (Fig. 3). P8 distinguishes between ‘transition’ - “…the switch of the responsibility of the service from the client to the service provider…”, and ‘transformation’ - that reflects the internal processes of the service provider to “…integrate the new process to the existing procedures and standards”.

Third, the ‘mini due diligence’ stage could perhaps be relabelled. P8 noted that the investigated organization uses the term ‘due diligence’ as ‘mini’ may imply a short time phase or lack of significance (Fig. 3).

B. Research Question 2: What are the CSFs for IT outsourcing performance in the German automotive industry?

As already noted, the systematic literature review indicated possible critical success factors for the outsourcing process (Table I). The interviews afforded the opportunity to test out and/or validate these initial findings, and also to explore and develop new CSFs. The interviews by and large confirmed the initially developed CSFs but also generated some new ones. The eight interviewees mentioned a total of 79 CSFs.

Interview analysis has also reinforced the linking of specific CSFs to particular stages in the emerging process model; and some CSFs were clearly viewed as more significant than others. ‘Proper communication and information’ (CSF4), for example, was named by all participating persons. P2 suggested a development of this to introduce a transparent escalation structure, based on role and seniority, with the aim of protecting subordinate structures. New CSFs include the ‘culture of trust’ between the outsourcing partner and the outsourcer. In this context, P2 suggested team building activities to develop a feeling of solidarity and appreciation on both sides.

C. Research Question 3: Can a new process model be constructed for IT outsourcing in the German automotive industry that helps ensure quality outcomes and deliverables?

The main expected contribution of this research is to propose and develop a new conceptual process model for IT outsourcing in the German automotive industry. The contribution to knowledge builds on the exploration of the key steps and stages and their associated core activities, results and deliverables, and the relationship between them in IT outsourcing projects; and on an examination of the provisional CSFs and their development into a more concrete set of CFSs for maximizing IT outsourcing project quality outcomes. This is being achieved by studying the relationships between CSFs in different outsourcing stages, defining points of interconnection and establishing ways in which the process model can be used to improve quality outcomes and thereby reduce IT outsourcing project failure rates.

Taking into account the specific needs of IT outsourcing, and building on the provisional conceptual model, a generic overall process model can be constructed comprising nine stages: 1. Corporate Strategy, 2. Preparation, 3. Business Case / Tender, 4. First selection / due diligence, 5. Contract negotiation, 6. Transition, 7. Transformation, 8. Transfer of responsibilities / regular operation & innovation, 9. Monitoring & Control / In-Sourcing.

The mapping of the CSFs onto these stages in the process accommodates a significant number of new CSFs that were identified in the interviews (Fig 3). These are classified as "confirmation (c)" or "new (n)" CSFs in Table II, with provisional revised numbering shown in the right hand column. These have been applied to stages one and two in the process model as an example of the possible linkage between CSFs and project stages (Table II).

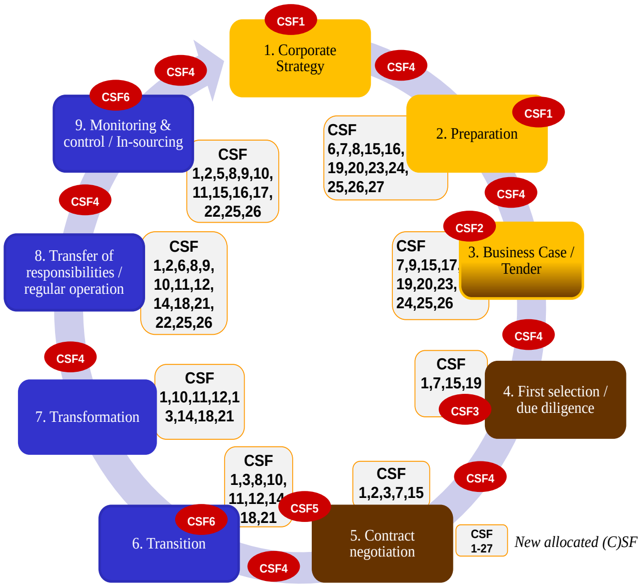


Figure 3. Nine-stage Process Model and (C)SFs

Transparency was highlighted as a key factor. One respondent suggested that the existence of different mental models (mind sets) of stakeholders in IT outsourcing projects occurs out of a lack of transparency. Other major issues highlighted were proper communication and the availability of information regarding strategic alignments. Respondents indicated as many as 15 CSFs within the preparation stage. CSFs in this stage mostly concern the quality of the tender (e.g. by understanding the customer requirements) and the early definition of fall back scenarios if the outsourcing fails.

Respondents identified the standardization of documentation and harmonization of services as one of the main factors that can improve the quality of the service in stage 3 (business case development and tendering). This is reflected in CSF 19, which proposes an evaluation matrix for a transparent and comprehensible selection process of the service provider. With regards to stage 4 (the first selection / due diligence), this CSF also has a major bearing as both sides (client and service provider) get to know each other’s views and positions for the first time.

Contract negotiation (stage 5) is characterized primarily by two tendencies epitomized in specific CSFs: transparency of operations and clarity of understanding of the substance of the outsourcing detail and organizational conditions. Integration of the procurement function as part of the overall outsourcing process is seen to have significant administrative and synergetic benefits.

For transition (stage 6), respondents highlighted CSFs with a strong human resource element e.g. “…encompass personal change processes of affected people…” and “culture of trust on both sides”. Participant responses suggested that interpersonal relationships constituted the largest risk of failure in this stage. The same applies to the new stage - transformation (stage 7): the service provider integrates the new processes in their work routine including the definition of the interfaces with customer personnel (be they retained by the customer or acquired by the provider).

Stage 8 focuses on the regular operation of, and innovation within, the new outsourced arrangement. Next to interpersonal factors, which play a critical role in this stage, the successful processing of new requirements and new technology developments during the contract period is a key contributor to deliver overall project success. The operation of the control model and the mapping of change requests into the tender/contract also surface as significant challenges. Stage 9 encompasses on-going monitoring and control activities as well as consideration of the option of in-sourcing. Clarity and transparency concerning strategic orientation (CSF 1a) is key for all stakeholders; and the business case has to be periodically revisited and questioned (CSF2). New service options may arise (e.g. offshore options), which were not an alternative at the start of the contract, and these need to be assessed. Transparency and clarity of communication and information (CSF4) are perceived as critical in this final stage of the outsourcing life-cycle.

Table II. CSFs for Stages 1 & 2 in the Conceptual Model

	n/c	Content	CSF
Stage 1	c	• Transparency concerning: ○ the strategic orientation ○ internal IT performance ○ customer requirements	CSF 1 CSF 1a CSF 1b CSF 1c
	c	• Proper communication and information	CSF 4
Stage 2 Preparation	c	• Transparency concerning: ○ the strategic orientation ○ customer requirements	CSF 1 CSF 1a CSF 1c
	c	• Economic point of view: value for money (management of costs)	CSF2
	c	• Proper communication and information	CSF 4
	n	• Service control and lessons learned (e.g. define service level agreements, documentation, etc.)	CSF 6
	n	• Include all knowledge carriers to improve the quality of the tender (e.g. controlling for the business case calculation)	CSF 7
	n	• Accompany the personal change processes of the affected people	CSF 8
	n	• Clear understanding concerning the support services of the procurement department (in scope / out of scope)	CSF 15
	n	• Early involvement of involved departments, e.g. the procurement department as a basis for bundling activities and achieve synergies	CSF 16
	n	• Development of a catalogue of criteria including qualitative and quantitative issues to evaluate the tenders and service providers	CSF 19
	n	• Tender documents comply with current standards, guidelines and criteria of the company.	CSF 20
	n	• Consideration of exit strategies	CSF 23
	n	• Definition of a fluctuation rate by the contracting authority	CSF 24
	n	• Clear definition of a control model and their mapping in the tender	CSF 25
	n	• Handling of new requirements and new technologies during the contract period	CSF 26
	n	• At the project level: definition of fall back scenarios, if the outsourcing fails	CSF27

V. Conclusions and Future Work

The process model developed to date has built upon existing literature, and initial findings have been developed and verified through detailed interviews with outsourcing participants. This model is now viewed as a suitable basis for further analysis of each of its nine stages, in which sub-processes can be identified, allied to further definition and development of CSFs. This allows the development of templates and guidance materials to aid practitioners in the IT outsourcing process in the automotive industry.

Interviewee analysis has supported a new definition of IT outsourcing. Despite differences in perception of what the concept is, there is some common ground in all interviewees' understanding, notably that the concept involves a transfer of a defined set of tasks to an external service provider. After a process of identification of key themes, and feeding back options to participants, the 'best fit' definition is that "IT outsourcing is the transitioning of IT services to an outside vendor with the aim of creating value for customers and providing services based on a previously defined sourcing strategy and clearly formulated core competencies, encompassing budget and headcount issues, risk management, future control and communication processes". As regards the large number of CSFs that surfaced in the initial interviews, an onion model to validate the importance of named CSFs within the interviews was developed. This combines the frequency a single factor was named by the participants with the frequency of occurrence within the life-cycle. This has allowed the development of hierarchical tiers of success factors, with tier 1 being most important and tier 5 least important (Fig.4 and Table II), and only tiers 1 and 2 being deemed critical as discussed by Delmour [33]. Tiers 3 to 5 are seen as important influencing factors but not critical to project success, productivity and sustainability.

As one of the visionaries of recent times in the IT and business field has noted, "there is no relationship between expenses for computers and business profitability…..You

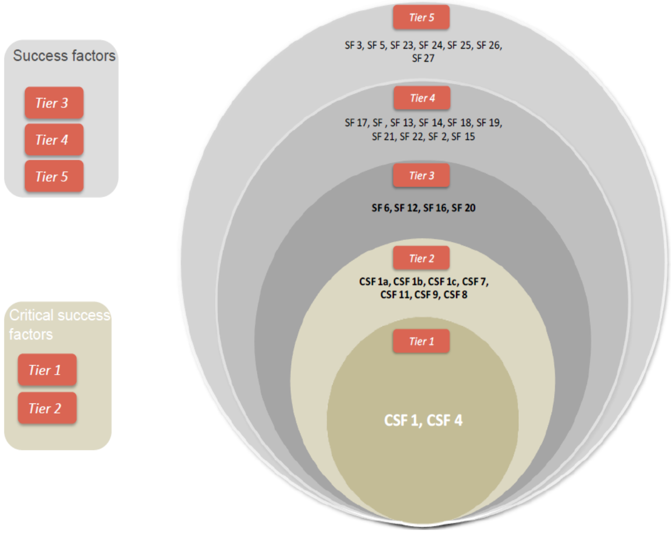

Success factors
Tier 3
Tier 4
Tier 5
Critical success factors
Tier 1
Tier 2
Tier 5
SF 3, SF 5, SF 23, SF 24, SF 25, SF 26, SF 27
Tier 4
SF 17, SF , SF 13, SF 14, SF 18, SF 19, SF 21, SF 22, SF 2, SF 15
Tier 3
SF 6, SF 12, SF 16, SF 20
Tier 2
CSF 1a, CSF 1b, CSF 1c, CSF 7, CSF 11, CSF 9, CSF 8
Tier 1
CSF 1, CSF 4

Figure 4. (C)SF Tier Model

will find that similar computer technologies can lead either to monumental success or to dismal failures.”

Paul Strassmann [34] highlighted the importance of the management of the IT function and related processes in determining success or failure. This is particularly true of outsourcing operations, and the history of IT arguably evidences more outsourcing failures than successes. This research attempts to contribute to improving the outcome of IT outsourcing in the automobile industry. Results to date provide the platform for the future development of a range of learning materials to guide practitioners working in this field.

ACKNOWLEDGMENT

The authors acknowledge the support of AUDI AG in facilitating this research project and thank the participating interviewees for their contribution to the research findings.

REFERENCES

[1] S. Schäfer, IT Outsourcing Leitfaden: Von der Idee bis zur Umsetzung. Mit Praxisbeispielen und umfangreichen Checklisten: Books on Demand, 2011.

[2] E. Turban, E. McLean, and J. Wetherbe, Information Technology for Management Improving Quality and Productivity: Internet Supplement: John Wiley & Sons Canada, Limited, 1996.

[3] J. Goo, R. Kishore, and H. R. Rao., "A content-analytic longitudinal study of the drivers for information technology and systems outsourcing," presented at the Proceedings of the twenty first international conference on Information systems, Brisbane, Queensland, Australia, 2000.

[4] F. Duhamel, I. Gutiérrez-Martínez, S. Picazo-Vela, and L.F. Luna-Reyes (2012). The Key Role of Interfaces in IT Outsourcing Relationships. 5 (1), 37-56.

[5] E. Groetschel. (2006, 2006 Jan 19th) Warum Outsourcing-Projekte scheitern. Computerwoche. Available: http://www.computerwoche.de/management/it-services/571060/

[6] J. R. Hofmann, J.H. Werner Schmidt, and W. Renninger, Masterkurs IT-Management: Grundlagen, Umsetzung Und Erfolgreiche Praxis F R Studenten Und Praktiker: Vieweg+teubner Verlag, 2010.

[7] Z. Süddeutsche, "Audi bei IT angriffslustig " in Süddeutsche Zeitung, ed. Munich, 2012.

[8] A. Gopal and S. Gosain, "Research Note—The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance," Information Systems Research, vol. 21, pp. 960-982, 2010.

[9] D. Eschlbeck. (2009, March 15th, 2011). Die Auswirkungen von Outsourcing im IT-Bereich auf unternehmerische und räumliche Strukturen. Available: http://books.google.de/books?id=wNY8Y-fg12gC&printsec=frontcover&dq=Die+Auswirkungen+von+Outsourcing+im+ITBereich+auf+unternehmerische+und+r%C3%A4umliche+Strukturen&source=bl&ots=0OWNNYdsKi&sig=AufHhgS51BrQJnJtnhScq3oFpIA&hl=de&ei=YweBTffrGs7Lsgb9pb3kBg&sa=X&oi=book_result&ct=result&resnum=2&ved=0CDQQ6AEwAQ#v=onepage&q&f=false

[10] L. Schwarze and P. Müller, "IT-Outsourcing - Erfahrungen, Status und zukünftige Herausforderungen," HMD 245, vol.10 , 42. Jahrgang, 2005, pp. 6-17.

[11] S. Cullen, Seddon, P., and Willcocks L. (2005, March 10th, 2011). Managing outsourcing: The life cycle imperative. 4. Available: http://infosys.uncc.edu/mbas6320/Readings/managing%20outsourcing.pdf

[12] R. Gonzalez, J.Gasco, and J. Llopis, "Information Systems outsourcing: An empirical study of success factors," Human Systems Management, vol. 29, 2010, pp. 139-151.

[13] R. Gonzalez, J.Gasco, and J. Llopis, "Information systems outsourcing success factors: a review and some results," Information Management & Computer Security, vol. 13, 2005, pp. 399-418.

[14] J. M. Callahan, "10 practical tips for successful outsourcing," hfm (Healthcare Financial Management), vol. 59, 2005, pp. 110-116.

[15] G. Petter and S.-S. Hans, "Critical success factors from IT outsourcing theories: an empirical study," Industrial Management & Data Systems, vol. 1052005, pp. 685-702,.

[16] M. C. Lacity, S. Khan, A.H. Yan, and L.P. Willcocks, "A review of the IT outsourcing empirical literature and future research directions," Journal of Information Technology, vol. 25, Dec. 2010, pp. 395-433.

[17] O. J. Akomode, B. Lees, and C. Irgens, "Constructing customised models and providing information for IT outsourcing decisions. Logistics Information Management," vol. 11, 1998, pp. 114-127.

[18] H. Solli-Saether and P. Gottschalk, "Maturity in IT outsourcing relationships: an exploratory study of client companies," Industrial Management & Data Systems, vol. 108, 2008, pp. 635-649.

[19] G. Petter and S.-S. Hans, "Critical success factors from IT outsourcing theories: an empirical study," Industrial Management & Data Systems, vol. 105, 2005, pp. 685-702.

[20] J. Goo, "Structure of service level agreements (SLA) in IT outsourcing: The construct and its measurement," Information Systems Frontiers, vol. 12, Apr. 2010, pp. 185-205.

[21] J. Goo and C. D. Huang, "Facilitating relational governance through service level agreements in IT outsourcing: An application of the commitment-trust theory," Decision Support Systems, vol. 46, Dec. 2008, pp. 216-232.

[22] J. A. Maxwell, "Qualitative Research Design: A Interactive Approach," vol. 2nd Edition, C. S. P. Thousands Oaks, Inc., Ed., ed, 2005.

[23] R. Audi, The Cambridge dictionary of philosophy. Cambridge ; New York: Cambridge University Press, 1995.

[24] K. Nogeste, "Research Strategy Development for Dummies: Define a Framework for Research and than use it,," presented at the European Conference on Research Methodology for Business and Management Studies, Lisbon, Portugal, 2007.

[25] M. Saunders, P. Lewis, and A. Thornhill, Research methods for business students, 5th ed. New York: Prentice Hall, 2009.

[26] J. W. Creswell, Research design: qualitative, quantitative, and mixed method approaches: Sage Publications, 2003.

[27] U. Flick, An introduction to qualitative research: SAGE, 2009.

[28] R. K. Yin, Case Study Research - Design and Methods, 4 ed. Thousand Oaks: Sage, 2009.

[29] E. Taylor-Powel and, M. Renner, Analyzing Qualitative Data: University of Wisconsin--Extension, Cooperative Extension, 2003.

[30] A. Funk, T. Söbbing, S. Neuhaus, W. Fritzemeyer, J. Schrey, and R. Niedermeier, Handbuch IT-Outsourcing: Recht, Strategie, Prozesse, IT, Steuern samt Business Process Outsourcing: Müller Jur.Vlg.C.F., 2006.

[31] H. Smuts, A. van der Merwe, P. Kotzé, and M. Look, "Critical success factors for information systems outsourcing management: a software development lifecycle view," in SAICSIT '10 Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, ed. Bela Bela, South Africa, 2010, pp. 304-313.

[32] A. Ishizaka and R. Blakiston, "The 18C's model for a successful long-term outsourcing arrangement," Industrial Marketing Management, vol. 41, 2012, pp. 1071-1080.

[33] B. Dellmour, Critical success factors it-start-up companies: GRIN Verlag, 2011.

[34] Paul A. Strassmann, The Business Value of Computers, The Information Economic Press, New Canaan, Conneticut, 1990.

Web Science Studies into Semantic Web Service-Based Research Environments

Mark D. Wilkinson
Centro de Biotecnología y Genómica de Plantas
Universidad Politécnica de Madrid (UPM)
Madrid, Spain
mark.wilkinson@upm.es

Abstract— **The emergent domain of Web Science has a number of as-yet unrealized goals. Among these are: to facilitate scientific discourse by supporting the explicit comparison and evaluation of hypotheses; to simplify *in silico* experiments by providing an ecosystem of expert analytical strategies that can be automatically assembled; to enhance scientific rigor by reducing bias, and improving reproducibility; and to integrate the knowledge gained from the experiment back into the Web. SHARE is a novel orchestration system that automatically chains-together Semantic Automated Discovery and Integration (SADI)–style Semantic Web Services. During development of SHARE, we noted that many requirements of such an end-to-end Web Science research environment were being realized. These include formally-defined, machine-readable, Web-embedded research hypotheses; an explicit, transparent, rigorous, and reproducible research methodology utilizing the most up-to-date data and expert-knowledge from the community; immediate dissemination and re-use of the resulting data and knowledge; and enhanced support for peer-review. This manuscript describes how SHARE is now being tested as a prototype Web Science framework.**

Keywords – SADI, SHARE, Semantic Web Services, Workflow Orchestration, Reproducibility, Transparency, Personalized Web

I. MOTIVATION AND REQUIREMENTS-GATHERING

Web Science is a recently-established, cross-disciplinary research domain spanning technology, sociology, psychology, and policy. Web Science research considers the Web as both a subject of, and/or a platform for, scientific investigation. It is the latter perspective that is the focus of this work. This domain of Web Science includes investigations into how Web environments might augment many aspects of scientific research, from scientific discourse through experimentation to peer-review and publication. In this manuscript, we are interested in how Semantic Web technologies might improve the execution of, and reproducibility of, high-throughput biomedical research.

Arguably, the Web is the most important and ubiquitous tool in modern biological research; used to collect data, submit data for processing, research prior art, and publish research results, there are few moments in the day when a biological researcher does not have their Web browser open. One could therefore argue that we are already engaged in "Web Science". However, the Web, as used by researchers today, is merely a conduit through which ideas and results are transmitted, usually in a form that the Web itself cannot “understand” or take advantage of (i.e. PDF-formatted discussions); like a carrier pigeon delivering paper notes, current Web Science is done *over* the web, not *within* it. This significantly under-utilizes the power of modern Web technologies – Semantic Web technologies in particular – to automatically discover and integrate data, knowledge, and analytical resources on a global scale. Such technologies are, therefore, the obvious choice for investigating novel Web Science infrastructures to support both the scientist and their science. The desire to understand if and how Web Science technologies can be applied is being driven by a number of intersecting trends, technical problems, and changing requirements in light of increasingly large datasets.

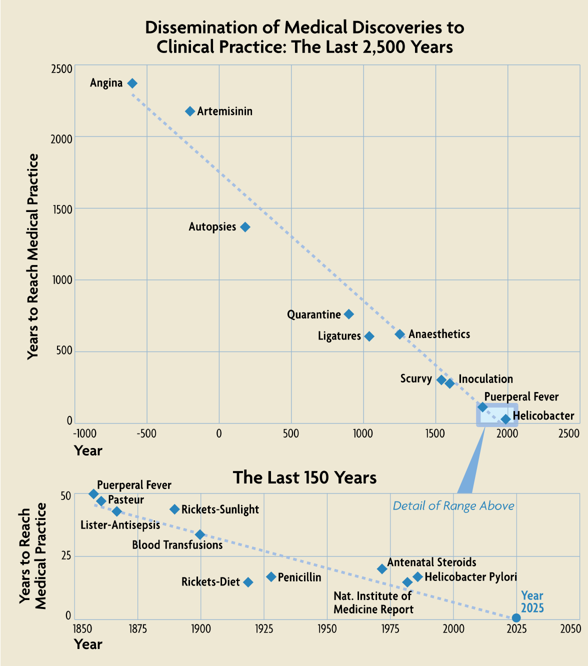


Figure 1. The time from discovery to implementation in Medicine. Over time, the delay between discovery and implementation has decreased in a near-linear manner. This pattern converges on the X-axis in approximately 2025 taken from [1] under c.c. License)

In Figure 1, Gilliam et al. suggest that the length of time between discovery, and the implementation of that discovery in-practice, is shrinking rapidly [1]. The rate at which discovery and implementation are converging appears constant, and the two are predicted to meet in approximately 2025. At this point, according to Gilliam, the moment of discovery, dissemination, and utilization merge into a single event. While the figure relates to healthcare, the same phenomenon likely holds in all areas of life science.

In practice, to achieve this end-point, research would be conducted in a medium that immediately interpreted and disseminated the results of an experiment; disseminated these results in a form that immediately (and actively) affected the results of other studies; and affected those studies without requiring those investigators to be aware of the new results or knowledge (since researchers are unable to stay-abreast of the literature, even in their own specialized field [1]). Moreover, it would be desirable for the experiment to be thoroughly documented, in a machine-readable manner, with a full provenance record including purpose, hypothesis, source, the data and algorithms used, versions, etc. This would allow both machines and humans to better assess the reliability, applicability, and validity of these results prior to using them in subsequent experiments. Despite being projected to be only a decade away, such knowledge-generating and knowledge-disseminating technologies and frameworks do not yet exist.

In parallel with the growing requirement for speed in knowledge-dissemination, there are increasingly worrisome observations of human limitations with respect to managing and manipulating these massive and complex datasets, and the resulting ease of making errors during data analysis - "the most common errors are simple... the most simple errors are common" [2]. Many researchers lack the skills to programmatically manipulate large datasets, and continue to use inappropriate tools to manage 'big data'. Serious errors introduced during data manipulation are difficult to detect by the researcher and, because they go un-recorded, are nearly impossible to trace during peer-review. In addition, the statistical expertise required to correctly analyze high-throughput data is rare, and biological researchers are seldom adequately trained in appropriate statistical analyses of high-throughput datasets. As such, inappropriate approaches, including trial-and-error, may be applied until a "sensible" answer is found [3]. Finally, because manually-driven analyses of high-throughput data can be extremely time-consuming and monotonous, researchers will sometimes inappropriately use a hypothesis-driven approach – examining only possibilities that they already believe are likely, based on their interpretation of prior biological knowledge, or personal bias towards where they believe the "sensible" answer would be found [4]. Thus, the scientific literature becomes contaminated with errors resulting from "fishing for significance", from research bias, and even from outright mistakes. These problems are becoming pervasive in omics-scale science - the affordability and accessibility of high-throughput technologies is such that now even small groups and individual laboratories can generate datasets that far exceed their capacity, both curatorially and statistically, to accurately manipulate and evaluate.

Even more troubling is that peer-review is failing to catch serious errors. While the Baggerly study into high-throughput publication quality [2] triggered retractions and a scientific misconduct investigation [5], the Ioannidis study reveals that, even in the prestigious Nature Genetics, more than half of the peer-reviewed, high-throughput studies cannot be replicated [6]. The failure of peer-review to detect non-reproducible research is, at least in part, because the analytical methodology is not adequately described [6], but perhaps equally because a proper evaluation of an experiment that controlled for errors would necessitate a re-execution of the experiment itself – something that is not reasonable to expect from reviewers. Thus, in the "big data" world, traditional peer-review is demonstrably ineffective.

In recognition of these limitations, the Institute of Medicine in 2012 published several recommendations relating to proper conduct of high-throughput analyses [7]. These include: rigorously-described, annotated, and followed data management procedures; "locking down" the computational analysis pipeline once it has been selected; and publishing the workflow of this analytical pipeline in a formal manner, together with the full starting and result datasets. These recommendations help ensure that (a) errors are not introduced through manual data manipulation, (b) there can be no human-intervention in the data as it passes through the analytical process, and (c) that third-parties can properly evaluate the data, the analytical methodology, and the result at the time of peer-review. While formal workflow technologies have proven effective at resolving some of these issues [8], integration of workflows into the overall scientific process continues to be *ad hoc*, and workflows themselves tend to be sparsely documented, difficult to review, difficult to re-use and re-purpose, and are not integrated with other forms of Web knowledge and expertise [9].

We lack the frameworks, standards, and infrastructures required to meet most of the intersecting trends, requirements and recommendations described above; moreover, these requirements appear to necessitate mechanization of much of the scientific process. As such, there is now some urgency around the necessary and inevitable creation of next-generation Web Science technologies, frameworks, and infrastructures to support the activities of high-throughput researchers.

In the remainder of this work-in-progress manuscript, we first describe, in Section II, the results of our examination of a prototype Semantic Web Service-based Web Science platform. We then discuss, in Section III, the underlying technologies that were used in that prototype, and how we propose to further examine these technologies. We then conclude in Section IV with a brief discussion of related projects, and the potential impact such platforms might have on the scientific process.

II. A PROTOTYPE WEB SCIENCE RESEARCH PLATFORM

Almost ubiquitously, scientific Web Services exhibit a set of features/behaviors that make them easier to connect into workflows compared to business-oriented Services [10]. We leveraged this to create a prototype Semantic Web Service workflow orchestration engine, and the results of these studies were recently published [9]. We demonstrated that, by constructing and publishing in the Web a semantic model – an ontology – describing a hypothetical biological phenomenon (Figure 2, left), we were able to automatically synthesize and execute an integrative analytical workflow (Figure 2, right) that discovered and/or synthesized data matching that model. Put concisely, by building a formal

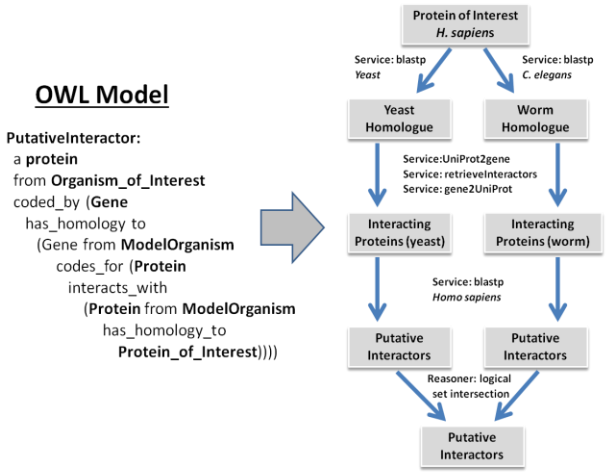


Figure 2. Conversion of models into analytical workflows: A biological model, in this case describing the chain of properties that would be expected of proteins that interact in a particular organism, is constructed in OWL (left). The SHARE software then analyses the knowledge in this model, and constructs an analytical workflow of inter-connected SADI Semantic Web Services (right) which is capable of generating data that conforms to that model.

model representing a hypothetical biological scenario, we were able to automatically find data compatible with that scenario without introducing (manual) bias. Moreover, the hypothetical model, the analytical workflow, and the result, were: (a) explicit and machine-readable; (b) inherently connected into local and remote data-sets on the Web, (c) a merger of explicit local and remote biological data, knowledge and analytical expertise, and (d) automatically published on the Web for peer-review and re-use.

III. TECHNOLOGIES AND APPROACH

The Web Science research platform above utilized both standard Semantic Web technologies, as well as novel tools developed in our group. The core technologies were:

Resource Description Framework (RDF) - a data syntax consisting of statements in the form of "triples" of subject, predicate, and object, where each component of the triple is a Uniform Resource Identifier (URI).

Web Ontology Language (OWL) - a description logic used to create machine-readable assertions that direct the automated interpretation of sets of RDF triples. It is worth noting that this use of OWL – for *ad hoc* data interpretation – is not typical within the Life Sciences community, where OWL is more often used to create concept hierarchies, or as a data template or schema. A Web Science research platform, however, requires discovery of data matching a hypothetical model from a 'mashup' of multiple disparate external data resources that will not have a predictable structure. As such, biological knowledge representing hypotheses is modeled in OWL, and this model is used to both formulate the experimental workflow (see "SHARE" below) as well as to automatically discover matching data hidden within vast integrated datasets.

Semantic Automated Discovery and Integration (SADI) [11] is our set of design patterns for scientific Semantic Web Service publishing. SADI Services are distinct in that they require Web Service publishers to (a) consume and produce RDF natively; (b) model their input and output as OWL classes; and (c) explicitly model the semantic relationship between input and output data through properties in the output class. The SADI design patterns dramatically improve the ability of software to automatically discover appropriate data retrieval and analysis services, and chain these into complex analytical workflows [11].

Semantic Health and Research Environment (SHARE) [12] is a specialized SPARQL-DL query engine that (a) responds to SPARQL queries by mapping query clauses to SADI Semantic Web Services, and (b) finds instances of an OWL ontological class by recursively mapping the class-defining property restrictions to SADI service invocations, then pipelining those services into an automatically-executed workflow. Succinctly, given the OWL model of a biological concept, it will attempt to find data on the Web consistent with that model. What is most important to note about SHARE is that is capable of automatically mapping a biological model onto a computational workflow made-up of SADI services. Since our belief is that a Web Science research platform should automatically evaluate arbitrary biological hypotheses, it is this separation of, and automated mapping between, the formal biological question, and the formal computational solution, that led us investigate SHARE's utility and behaviors further. In particular, the Web Science-like features of the SHARE *in silico* research platform include:

1) That the research process is entirely Web-embedded
2) Distributed expert-knowledge encoded in OWL is used, through OWL imports, both to construct the hypothesis as well as to drive the formulation of the solution; thus, a researcher need not possess all of the knowledge the experiment requires.
3) The experimental workflow is explicit, and no manual intervention occurs during the execution of the experiment.
4) The experimental workflow can be re-generated and re-run over the same dataset (reproducibility) and more importantly, will automatically adapt to a new dataset (re-usability) due to SHARE's ability to dynamically discover appropriate Services based on both the specific dataset and the OWL model [9].
5) The experimental workflow is self-annotating, as a result of being derived from a biological model; thus, review of the experiment is dramatically simplified with no additional human curation.
6) The starting datasets, and result datasets, are encoded in RDF, and thus are by nature a part of the Web. These, together with the initiating OWL model and workflow, aid third-party evaluation.

To investigate this Web Science platform further, we are now attempting to determine the extent to which common high-throughput life science problems can be modeled in OWL, and test the resiliency of the SHARE OWL-to-workflow transformations. In particular, the limitations of the OWL/OWL-2 languages are well-known, and we anticipate that certain hypothetical constructs will require other types of semantics or logical filters, such as rules and/or the filters available within the SPARQL language itself. To explore the boundaries of what types of *in silico* hypotheses can be modeled using available logics and rules languages, the following experiments are being conducted:

1) Select 15-20 peer-reviewed papers spanning a broad range of high-throughput experimental scenarios, ensuring that the datasets and Services required to execute the experiment are available.
2) Empirically attempt to model these in OWL.
3) For any hypothetical construct asserted in the publication that cannot be modeled in OWL, attempt to model it using an alternative tool such as rules or using query filters.
4) For any construct that cannot be modeled using any available tool or language, construct an appropriate Web Service that will generate data conforming to this construct (while we acknowledge that building a Web Service to resolve every difficult case is not a scalable solution, it ensures that we can progress to the end of the investigation).
5) Manually construct a workflow, in collaboration with biological researchers, that represents their expert solution to the hypothesis.
6) Provide the OWL/rules model to SHARE; examine and compare the workflow that is automatically constructed.
7) Similarly, compare the outputs of the two workflows to determine if any methodological differences were consequential. Biologists will also evaluate the automatically-generated output.

IV. Discussion and Conclusion

Approaches to research methodology and scientific publishing have changed very little with the advent of the Web, and remain largely unaffected by the emergence of powerful Semantic Web standards and tools. We believe that, by continuing to develop the technologies described here, we can dramatically change the way *in silico* research is conducted and disseminated by establishing the fundamental concepts of “Web Science” – a novel approach to scientific research where hypotheses and experiments are explicitly described, publicly shared, intimately linked into existing data and knowledge, dynamically executed in an unbiased manner using the encoded expertise of the global community, and automatically published with a full provenance record, enabling rigorous peer-review. We propose, further, that formal semantic models can (and should) be used as the mediators of scientific discourse and disagreement.

The objective with this report is to present these ideas to the community in order to: raise awareness of the project; foster debate about its plausibility and utility; and encourage both collaborative and independent pursuit of similar research problems with the goal of rapidly bringing Web Science to fruition. In this regard, we are currently collaborating with the HyQue project [13] which utilizes RDF/OWL to model data and knowledge, and evaluates hypotheses formulated as SPARQL queries using a novel ontology. The significant differences are that SHARE uses distributed resources rather than a warehouse; utilizes Web Services, thus can execute analyses in addition to static data retrievals; and does not utilize any project-specific ontologies. Nevertheless, these independently-derived, yet highly similar, Web Science research environments suggest that the potential “solution space” for Web Science infrastructures may be very small.

Acknowledgment

Funding from the Marie Curie Co-fund, the Heart and Stroke Foundation of Canada, Microsoft Research, the Canadian Institutes for Health Research, and CANARIE.

References

[1] M. Gillam, et al. “The healthcare singularity and the age of semantic medicine,” in *The Fourth Paradigm: Data-Intensive Scientific Discovery*, T. Hey, S. Tansley, and K. Tolle, Eds. Microsoft Research, 2009, pp. 57–64.

[2] K. A. Baggerly and K. R. Coombes, “Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology,” *Ann. Appl. Stat.*, vol. 3, no. 4, Dec. 2009, pp. 1309–1334.

[3] A.-L. Boulesteix, “Over-optimism in bioinformatics research” *Bioinformatics*, vol. 26, no. 3, Feb. 2010, pp. 437–439.

[4] P. Fisher, et al., “A systematic strategy for large-scale analysis of genotype phenotype correlations: identification of candidate genes involved in African trypanosomiasis” *Nucleic Acids Res.*, vol. 35, no. 16, Jan. 2007, pp. 5625–33.

[5] V. Gewin, “Research: Uncovering misconduct,” *Nature*, vol. 485, no. 7396, May 2012, pp. 137–139.

[6] J. P. A. Ioannidis, et al., “Repeatability of published microarray gene expression analyses” *Nat. Genetics*, vol. 41, no. 2, Feb. 2009, pp. 149–155.

[7] C. M. Micheel, S. J. Nass, and G. S. Omenn, Eds., *Evolution of Translational Omics Lessons Learned and the Path Forward*. The Institute of Medicine of the National Academies, 2012, p. 354.

[8] M. Hauder, Y. Gil, R. Sethi, Y. Liu, and H. Jo, “Making data analysis expertise broadly accessible through workflows” in *Proceedings of the 6th workshop on Workflows in support of large-scale science - WORKS ’11,* 2011, p. 77.

[9] I. Wood, B. Vandervalk, L. McCarthy, and M. Wilkinson, “OWL-DL Domain-Models as Abstract Workflows” in *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies*, T. Margaria and B. Steffen, Eds. Springer Berlin/Heidelberg, 2012, pp. 56–66.

[10] P. Lord, et al., “Applying Semantic Web Services to Bioinformatics: Experiences Gained, Lessons Learnt” *LNCS*, vol. 3298, Jan. 2004, pp. 350–364.

[11] M. D. Wilkinson, B. Vandervalk, and L. McCarthy, “The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation” *J. Biomed. Semant.*, vol. 2, no. 1, Oct. 2011, p. 8.

[12] B. Vandervalk, L. McCarthy, and M. Wilkinson, “SHARE: A Semantic Web Query Engine for Bioinformatics” in The Semantic Web, *LNCS Proc. ASWC 2009*, vol. 5926, pp. 367–369.

[13] A. Callahan, M. Dumontier, N.H. Shah, "HyQue: Evaluating hypotheses Using Semantic Web Technologies," *J. Biomed. Semant.* vol 2(Suppl.2): S3.

Versioning and Historiography in Automated Generic Electronic Flight Log Book Transfer

Arne Koschel, **Carsten Kleiner**
Univ. of Applied Sciences and Arts,
Faculty IV (Dept. of Computer Science)
Hannover, Germany
akoschel@acm.org, ckleiner@acm.org

Björn Koschel
edatasystems GmbH

Gelsenkirchen, Germany
bjoern.koschel@edatasystems.de

***Abstract*—The automated transfer of flight logbook information from aircrafts into aircraft maintenance systems leads to reduced ground and maintenance time and is thus desirable from an economical point of view. Until recently, flight logbooks have not been managed electronically in aircrafts or at least data transfer from aircraft to ground maintenance system has been executed manually, since only latest aircraft types (e.g., Airbus A380, Boeing 777) support electronic logbooks. This paper introduces a top level distributed system architecture of a generic system for automated flight logbook data transfer. The system includes a generic mapping component that facilitates flexible mappings between aircraft logbook systems as input and aircraft maintenance systems in the backend. As its main contribution this paper details versioning and historiography concepts for our system. The former makes it possible to deal with different versions of input and output systems in a single mapping component. The latter explains the practically important aspect of historizing data in order to comply with legal regulations. Due to its flexible design the mapping component could also be used for other domains with similar requirements.**

Keywords-System integration; versioning; historiography; aerospace domain; generic interface; flexible data mapping

I. Introduction

Ground and maintenance time is very costly for airline operators. They try to minimize them for economical reasons. Today's mostly manual transfer of flight logbook data from an aircraft into the operator's maintenance systems should be automated to get closer to this goal. This should reduce information transfer time and is likely to be less error prone as well. Thus, in total it should result in reduced ground maintenance time and cost.

A generic automated flight log data transfer system needs to support differently structured flight log information from different aircraft types and manufacturers on one side. On the other side different aircraft maintenance systems used by different operators have to be supported. Technically, all these systems are likely distributed, even though typically in a common network within a single organization. Moreover, fault tolerance and (transactional) persistence to prevent data loss are required. Not very critical are performance demands due to a limited amount of log data per flight in practise.

To support those requirements, a *generic* transfer system for flight logbooks into maintenance systems needs to be designed and implemented. In a joint industry and research cooperation (*Verbundprojekt*) the eLog system has been designed and prototypically implemented by the University of Applied Sciences Hannover in cooperation with Lufthansa Technik AG and edatasystems GmbH. Technically this system supports different – currently Extensible Markup Language (XML)-based, but with heterogeneous XML schemata – versions of different aircraft flight log systems on the *input* side. On the *output* side different airline systems are supported, which may have different data models. As an example a Relational DataBase Management System (RDBMS/DBMS) with tables is used, that map (almost) 1:1 to the data structures of the *output* system, which is Lufthansa's core aircraft maintenance system. Note that the number of potential input and output systems in a real world scenario will be rather small (e.g. a one-digit number), thus limiting the number of potential combinations of input and output systems and the mappings required.

The resulting eLog system offers a generic distributed system architecture for integration and mapping of the different XML-based flight log input data formats to different output aircraft maintenance systems. The mapping is configurable and flexible, including mapping of arbitrary entities.

Mapping of input objects to output objects is specified by an XML mapping document (conforming to a specific mapping schema) which is dynamically loaded into the mapping component (cf. sec. IV). Information in the mapping document is handled dynamically making the component extremely agile once in operation. In total all these features contribute towards our goal of a generic flight logbook tool. In [13], a high level overview on eLog has been given and [14] discusses the mapping component in detail. This paper's main contribution are details on the versioning and historiography concept for eLog. The versioning concept is important in order to deal with different versions of input and output systems with a minimal operational overhead. Thus, a single (or at least small number) of mapping system instances should be able to take care of several versions of input and output systems with each combination requiring individual mapping specifications. Historiography is important both from an operational point of view (e.g., to limit the amount of data to be held in the operational mapping systems) as well as from a legal perspective as certain accountability requirements for all Information Technology (IT) systems related to aircraft are in place.

Initial tests of the prototypical implementation already validated the practical usefulness of the concept. Very few logbook data transfer systems exist. Those that do are flight operator internal and/or aircraft type and maintenance system specific.

Although concepts and approaches for application/data integration in general of course do exist (important ones are briefly discussed in section II) applying them to flight logbook data is a novelty. To the best of our knowledge a *generic* flight logbook data transfer system has not been implemented yet and is thus the key contribution of our overall work.

The remainder of this article discusses some related work in section II before giving a high level system overview of the eLog system in section III that has been the result of an extensive comparison of options (cf. [13]). Section IV briefly introduces the generic mapping specification before section V discusses as the main contribution the eLog versioning and historiography approach. The article ends with a conclusion and some outlook to future work in section VI.

II. Related Work

Related work originates from different areas. *Conceptually* different enterprise application integration approaches [2], [4], [12] provide a potential foundation for the technical system architecture. The most important ones include: messaging based enterprise integration patterns [7], transactional DBMS based approaches [5], [3], using an Enterprise Service Bus (ESB)/Service-Oriented Architecture (SOA) as foundation [11], and finally an Extract Transform Load (ETL)-like data warehouse concept [8].

All of them potentially deliver feasible architectures for a system like eLog and have been evaluated (cf. [13]). Eventually a transactional DBMS based architecture was chosen similar to an ETL approach. We omit a detailed discussion about the architecture selection at this point. The selection has been undertaken and is described in project internal documentation. It might be published by us in a future article. In this paper we will rather focus on the achieved results.

From an *application* point of view many systems exist, which transfer and map data from multiple input sources to different output sinks. For example, Deutsche Post uses an ESB for XML-based data transfer within a SOA [6].

A generic XML mapping architecture is discussed in [9]. Similarly [19] presents an algorithm for mapping of XML Schemas which we did not use to potentially missing XML schemats on output side. Graph based mapping to transform structured documents is explored in [16]. A tool for semantic-driven creation of complex XML mappings is presented in [17] whereas [20] presents a similar mapping approach to ours but employs XSLT for the mapping step. Moreover (semi-)automated mapping of XML schemas has been discussed in many research papers, a pretty comprehensive survey is given in [15]. However, for the limited complexity of the XML schemas used in flight logbooks the project has decided that the overhead of employing a complex automated mapper, let alone choosing the most suitable one, is too big. Nevertheless the output of an automated mapper could be used as a base to define the mapping documents (cf. section IV) if our approach is applied to more complex domain schemas. Also [18] describes the design of a histoy database but is restricted to a completely different domain.

Looking at recent aircraft models in particular shows that only the latest models support electronic log books. In addition standardization of the data format is still in its early stages. Recently initial standardization has been designed in the ATA specification [1]. This specification is quite helpful for the flight log input data within our work – although it is still significantly in flux. One e-logbook tool for the aerospace industry is presented in [10]. But no *generic* flight logbook data transfer system has been documented yet.

III. eLog: System Overview

The designed generic flight log data transfer system is based on ideas frequently found in ETL processes in data warehouse systems. Note though that the implementation itself is not based on data warehouses but rather uses a transactional DBMS-based approach as stated in the previous section. In order to provide a brief eLog system overview we explain the data flow within eLog in the sequel.

A. Data flow within eLog

Data flow within eLog follows a sequence of steps to map the XML input data to arbitrary aircraft maintenance systems. Figure 1 shows the high level input data flow from the flight logbook system until procured to the maintenance system.

An input reader component – where different implementations for different source data formats may exist – uses polling (step 1) to check whether new XML files have been delivered by aircrafts. Polling is implemented by frequently checking a predefined directory on a dedicated server for newly added files. The data is validated against the XML schema (step 2) of the particular aircraft's electronic logbook format, e.g., against those defined in [1]. If the data is formally valid, it is transferred into a buffer database (step 3), where it is stored in its original format in an *XML-type* attribute. Else an error handling takes place. As long as the covered aircrafts

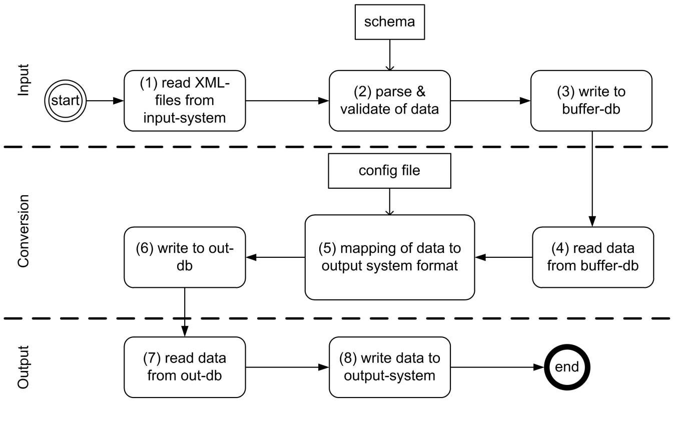

schema
Input
start
(1) read XML-files from input-system
(2) parse & validate of data
(3) write to buffer-db
config file
Conversion
(6) write to out-db
(5) mapping of data to output system format
(4) read data from buffer-db
Output
(7) read data from out-db
(8) write data to output-system
end

Fig. 1. Generic eLog data flow

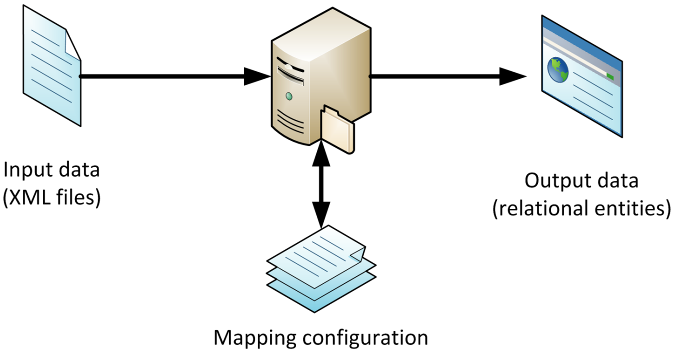


Fig. 2. XML based mapping process

provide source data in XML format a single database schema is sufficient for the buffer database.

Again using polling a mapping component checks for newly arrived source data (step 4) in the buffer database. It utilizes a flexibly configurable sequence of conversion functions to map the input data to a specific output target system (step 5); this step is explained in detail in section IV. The output data is stored in a database again (step 6). It closely resembles the data structure of the airline's maintenance system.

Consequently for each different maintenance system there will be an individual schema in the output database. Options in the mapping configuration include checks for dependent source information, flexible mapping and features to update existing entities in the output system; cf. details section IV.

Eventually another upload component transfers data from the output database (step 7) into the airline maintenance system (step 8) whenever an entity of the maintenance system has been completely assembled. As one option the airline maintenance system used provided a Web services interface for programmatic access.

IV. Mapping Rules

The mapping of domain specific information between input and output systems is expressed by XML mapping files conforming to a specific mapping schema. There is an individual mapping file for each entity of any input system that is triggering the creation of an entity in the output system (cf. fig. 2). Mapping files can take care of any type of mapping between input and output on entity, attribute and attribute value level. Mappings on entity level use an individual mapping specification for each of the input entity types. Within these entities nested elements specify the attribute mappings by individual converters whereas transformations on attribute values are defined on the innermost level within the target attribute specification.

For eLog it is important to note that there is an individual mapping configuration for each combination of input and output system where a mapping has to take place. This includes different versions of a system on each side, i.e., mapping from a different version of the same input system leads to a new mapping configuration file to be used. This has the advantage from an operational point of view that each combination and version of input and output system can be configured and operated independently of other potential mappings.

In the case study an implementation of the whole mapping process in Java has been performed. Technically the system is designed according to the ETL paradigm and it is implemented with different Java processes which together provide the tasks from figure 1. They are combined with a relational DBMS, that also allows for XML data storage (Oracle). Throughout the conversion steps DB transactions are utilized to ensure data consistency. In combination with operating system based fault tolerance (automated process restarts), a high degree of fault tolerance of the overall system is thus achieved.

The Java mapping application dynamically reads the mapping specification documents and checks for syntactical correctness. Thus, changes in the mapping specification simply require a restart of the mapping process without any changes to the source code. After starting, the transformation process polls the input folder (or database) for newly arrived XML input. Whenever the first input entity of a given type arrives, the mapping specification document is used to instantiate the required converters and functions as Java objects. The names of the converter XML elements are used to instantiate a corresponding Java class using reflection. This enables dynamic provisioning of converter classes and facilitates complex converters as the full power of the Java language can be used for implementation. Arguments of the converters and functions are provided to the Java classes based on the specific objects to be processed. Thus, additional converters and functions can be easily implemented and added to mapping processing by observing given interfaces without any changes to the mapping process source code. After successful mapping of an input entity the result is procured to the output system (a relational database in our case) with standard JDBC operations. Note that while the first step and the instantiation of mapping objects in the Java application are executed only once, the other steps are executed for each input entity with the number of executions depending on the particular mapping specification.

V. Versioning and Historiography

While the previous sections provided a brief general overview about eLog summarized from [13], [14], this section newly contributes approaches for two additional requirements, namely versioning and historiography within eLog.

Versioning within the eLog context in particular means to have independent version changes for the attached input systems such as ATA conformant systems as well as for the attached output airline maintenance systems. In principal arbitrary combinations of input and output system versions should be possible for eLog.

Historiography in the eLog context means what happens to the different input and output data within eLog's databases. Since this is mostly future work for eLog we only briefly sketch pragmatic ideas, which nevertheless might well form the basis for a pragmatic solution in this space.

This section will first examine the different resources within the eLog system architecture, which are affected by versioning.

It is followed by a look at some additional assumptions regarding foreseeable eLog usage. A discussion of several options to enable versioning for eLog based on the resources and the additional assumptions is performed. The section concludes with a look at eLog's historiography concept.

A. Versioning within eLog

1) Common resources in eLog affected by versioning: Taking a closer look at the eLog system architecture as sketched in figure 1 shows several resources, which are commonly used within eLog. Table I describes them in more detail.

2) Additional assumptions for versioning in eLog: Discussion with potential future users of eLog directed us to a few additional assumptions:

- There will be different versions of input data systems.
- There will be different versions of output data systems.
- Due to typical time and release schedules, the actual number of different versions on both sides – input and output – will be less then 10 per system.
- Individual administration is of importance on a 'per version' basis. Especially important is a very low impact of one system version (such as one particular Airman version on input side) to other system versions.
- A 'pragmatic' solution for version control with base technologies such as RDBMS and comparatively simple data structures is prefered by the proeject partners compared to an 'over-engineered' approach. For example, ontologies or version control systems are not required here.

3) Discussing input/output system versions for eLog: While the versioning discussion in principal is required for all of the mentioned system environments and common resources from table I, we will – due to space limitations – focus here on the versioning of the eLog database(s). Please note, that this nevertheless forms a baseline for a similar discussion of all those resources. For example, one could use a particular version of the eLog program 'tied to' a certain version of the eLog database or its users and schemas.

The discussion of versioning for the eLog database(s) is of particular importance, since it affects pretty much all other resources. We also do take into account for the discussion the 'additional assumptions' from above however. Based on an example we examine different approaches for versioning of the eLog database(s) below.

TABLE I
ELOG RESOURCES

System Environment	*Key Common Resources*
Physical or virtual machines to run the operating system	DB schema (user data, flight log entities, etc.)
Databases management systems such as Oracle 11	eLog program versions (and their respective run time process incarnations)
A certain runtime environment. For eLog Java virtual machines.	eLog data directories within the file system
	eLog configuration information

Typically a (relational) database installation will have a few database instances, which will have some schemas that will consist of some tables. For the following approaches let there be 2 input systems E,F with 3 version 0,1,2 as well as 2 output systems R,S also with 3 versions 0,1,2. From this example – which adheres to the 'additional assumptions' from above and would also be reasonably typical in practise for our project partner – the following versioning concepts arise:

- D1 – 1 DB instance per input/output system
  One full database instance per input/output system combination would allow for the best decoupling between different input/output system combinations. However, the price for many full DB instances is of course a relatively high overhead.
  Our example would result in 12 DB instances, one for each of the three versions of each of the four systems.
- D2 – 1 schema per input/output system and version
  In this concept only 1 database instance would be used. The separation between input/output system and resp. versions would be performed based on different DB users with their individual DB schemas.
  For the example this concept results in $\{E_0, E_1, E_2, F_0, F_1, F_2, R_0, R_1, R_2, S_0, S_1,$ $S_2\}$ thus 12 DB schemas within a single DB instance, one schema per input and output system and version.
- D3 – 1 DB schema per input/output system combination for all versions
  The number of different schemas could be reduced by using the same schema for all versions of a single input or output system. In this option all tables for a single system would be multiple for each version.
  For the example this gives $\{E_{0,1,2}, F_{0,1,2}, R_{0,1,2}, S_{0,1,2}\}$ thus 4 DB schemas with three times the tables each compared to D2. While the number of schemas is reduced significantly, the decoupling of input/output system combinations is much lower compared to D1 or D2 leading to increased dependencies during operation.
- D4 – 1 DB schema for all input systems and 1 DB schema for all output systems
  The lowest meaningful number of schemas would be just one for the data from all input systems and another one for all output systems.
  Here $\{(E, F), (R, S)\}$ would be the result for our example. This gives only two schemas, thus relatively low overhead but only minimal decoupling as the price.

A solution for a good relation between system / DB maintenance effort versus decoupling of systems may be found by looking at table II.

Option D2, which consists of 1 DB instance and 1 DB schema per input/output system and version provides a pragmatic solution with a suitable flexibility combined with a reasonable DB administration effort. It does require a substantial number of DB users – the sum of number of input systems and their versions plus number of output systems and their versions. However, the systems are decoupled. They can be

TABLE II
DB VERSIONING CONCEPT: EVALUATION

System	*Evaluation*	*Comment*
D1	–	Max. decoupling of versions, quite high overhead
D2	+	Maintenance effort vs. decoupling in a reasonable relation
D3	+-	Decoupling lesser, maintenance effort higher
D4	–	only 2 DB users required, but no decoupling for maintenance tasks given

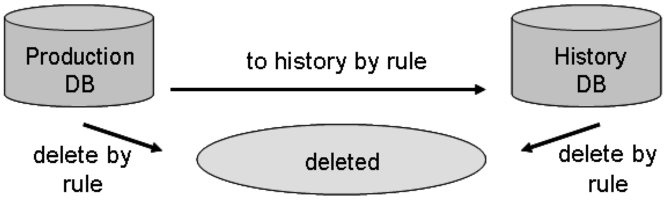


Fig. 3. Garbage Collection

maintained individually on a per version base, e.g., allowing for individual system starts and stops, backups or updates.

In total option D2 is thus the authors recommendation for eLog. However, individual airline IT environments, maintenance procedures or accounting strategies may have to lead to other options, e.g., to option D3. This would provide fewer flexibility, but it would limit the number of required DB users and thus reduce the DB administration effort to some degree.

Beside maintenance aspects the single point of failure aspect also has to be considered, at least, whenever system resources are shared. For this reason it is recommended for system resources, to have them separated whenever possible. For example, a shared DBMS as in option D2 might be used, but within a highly available environment. For separate eLog instances individual virtual or even physical machines shall be used. The number of parallel installations seems reasonable, while the advantages of independent administration, maintenance, and fault tolerance should be significant.

B. Historiography and 'Garbage Collection' for eLog

Persistent data within the eLog context resides in the buffer and in the output data base (cf. section III). Over time, the data stored there becomes outdated and thus – at least if in the future many aircrafts deliver input data for the system – the stored data wastes database memory in the eLog production databases. Still however, for control and accounting purposes outdated data should remain accessible in some form. For this reason some kind of data historiography and 'garbage collection' is a required feature for eLog. The base requirements for data historiography and garbage collection in eLog are thus:

- R1: Archive database
  Historiography and garbage collection in the eLog context means to archive data from the production databases into some other long term archive database according to certain rules (see below). When those rules apply the associated data must be moved from eLog's production database to an eLog archive database. Figure 3 sketches this process.
- R2: Rules for data historiography
  It shall be possible to declare certain (simple) time bound rules, which specify when and which data is moved to the archive database from R1 (and deleted afterwards).
- R3: Rules for data deletion
  Certain data items within eLog only serve for internal purposes, for example internal key and reference information for data mappings in eLog's output database. Once this information is outdated, it might safely be deleted ('garbage collected') with no need for further archiving. As in R2 it shall be possible to declare certain (simple) time bound rules, which specify when and which data might safely be deleted from eLog's production database.

Looking at those core requirements and the requirement of a pragmatic, established approach, the following solution seems sufficient: Conceptually each attribute gets two constraints (or markers). One specifies when it must be moved from the production database to the history database at the earliest (and deleted afterwards). The second specifies when the data item may be deleted from either production or history (whereever it resides) at the earliest. Please note, that from a regulatory persepctive may be a real deletion in the history table could even be 'never' or only in a very long time frame. At the moment, out project partners do not see this demand.

In table III fictive examples (although practically reasonable acc. to our project partners) are given for archiving and deleting certain flight log and maintenance data after a certain time period.

Technically again a pragmatic, proven solution is eLog's approach of choice. Periodic database jobs, which use triggers and stored procedures are a sufficient implemenation for eLog. External database programs, which run periodically, would be an alternative.

VI. CONCLUSION AND OUTLOOK

A. Conclusion

So far the concept and prototypical implementation of eLog have proven to be very promising. The implementation already covers a single exemplary input and also output system. It includes entities that require almost all possible mapping options between the systems. The mapping configuration as described above is defined in an XML file based on a proprietary schema.

TABLE III
TYPICAL ENTRIES: HISTORIOGRAPHY AND GARBAGE COLLECTION

Database	*Entity*	*toHistory after*	*toDelete after*
Buffer_DB	MaintLog	1 week	3 weeks
	MaintAction	1 week	3 weeks
	MaintRelease	no	3 weeks
Out_DB	mapping of MaintLog	no	3 weeks
	mapping of MaintAction	no	3 weeks
	mapping of MaintRelease	no	6 weeks

Different versions of input and output systems are an integral part of eLog's architectural concept (cf. section V). The mentioned XML schema may easily be adjusted to different input and output formats (for different versions of the same or other logbooks and maintenance systems). The mapping specification is read dynamically by the mapping component which makes easy and fast adjustments to different versions or products possible without any software development, just by configuration. The current output DB resembles the simple relational structure of most airline maintenance systems but may also be adjusted for a different maintenance system. Both mapping configuration as well as input and output data formats are sufficiently generic in order for the system to be easily adjusted to specific data formats. They might well be usable in other domains as well.

The overall modular design of the system by decoupling input and output system leads to a highly scalable overall architecture. Dealing with different versions and output systems will be a major issue as, e.g., logbook data will be procured to different output systems dealing with specific aspects of aircraft management. Since the actual number of different versions in an installation in a typical time period will in practise likely be small enough (in total likely less than 20), the scalability of the overall architecture will hold true.

Consequently, eLog's solution for versioning – as discussed in section V – is of highly practical importance for a working interface component. Moreover, the general discussion of versioning aspects could well be of practical importance for other systems as well.

Elog's pragmatic concept for data historiography based on history tables and (simple) rules could also be a solution for other systems with similar (relatively small) requirements.

B. Outlook

After a brief successful evaluation of the first eLog prototype we have finished the conceptual and detailed specification phase of eLog's 2^{nd} project stage and are about to finish this phase's implementation.

There the genericity of the mapping specification already proved to be quite useful as a different logbook provider was used than in the first stage. Also, the schema of the output system has changed from a relational schema towards a more XML-oriented schema using domain-specific objects from the aircraft maintenance domain. Moreover as of late two slightly different output systems will have to be procured positively testing the feasibility of our generic mapping approach as well as the versioning.

The next steps include evaluation of eLog in a production environment. All this will provide additional proof for the scalability and reliability aspects. Within the mapping specification we might also consider as future work some examination of ontology based approaches. Also, we will have to generalize the format of the mapping specification to an an XML schema in order to do development/compile time correctness checking of specifications in the future.

In summary, automated integration of flight logbooks shows the potential for reduced transfer times of maintenance information coupled with increased correctness when compared to the current manual process. This will eventually lead to reduced maintenance times for aircrafts and thus increase profitability of the airline. The presented real world eLog project is a very promising step into this direction.

Acknowledgment

The authors thank our cooperation partner Lufthansa Technik AG and our master students A. Hödicke and S. Nitz. The overall project is part of the Aviation Cluster Hamburg Metropolitan Region's Leading-Edge Cluster Strategy and is sponsored by the German Ministry of Education and Research.

References

[1] Air Transport Association of America, Inc. (ATA), "Industry Standard XML Schema for the Exchange of Electronic Logbook Data," 1.2 edition, May 2008.

[2] S. Conrad, W. Hasselbring, A. Koschel, and R. Tritsch, "Enterprise Application Integration," Spektrum, Germany, 2005.

[3] C.J. Date. "An Introduction to Database Systems, 7^{th} edt.," Addison-Wesley, U.S.A., 2003.

[4] J. Dunkel, A. Eberhart, S. Fischer, C. Kleiner, and A. Koschel, "Systemarchitekturen f. Verteilte Anwendungen," Hanser, Germany, 2008.

[5] R. Elmasri and S. Navathe, "Fundamentals of Database Systems, 5^{th} Edt.," Add.-Wes., U.S.A, 2006.

[6] M. Herr, U. Bath, and A. Koschel, "Implementation of a Service Oriented Architecture at Deutsche Post MAIL," In ECOWS, LNCS vol. 3250 Springer, 2004, pp. 227–238.

[7] G. Hohpe and B. Woolf, "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions," Addison-Wes., USA, 2003.

[8] W. H. Inmon, "Building the Data Warehouse," Wiley, U.S.A, 2005.

[9] W. Jiyi, "An Extensible XML Mapping Architecture," Chinese Control Conference (CCC2007), July 2007, pp. 291–293.

[10] A. T. Kavelaars, E. Bloom, R. Claus, et al, "An Extensible XML Mapping Architecture," In IEEE Transactions On Aerospace And Electronic Systems, 45(1), U.S.A, 2009.

[11] D. Krafzig, K. Banke, and D. Slama, "Enterprise SOA: Service Oriented Architecture Best Practices," Prentice Hall, U.S.A, 2005.

[12] D. S. Linthicum, "Enterprise Application Integration," Addison-Wesley, U.S.A, 1999.

[13] O. Hunte, C. Kleiner, U. Koch, A. Koschel, B. Koschel, and S. Nitz, "Automated generic integration of flight logbook data into aircraft maintenance systems," 17^{th} GI/ITG KiVS, OASIcs, Dagstuhl, Germany, 2011, pp. 201–204.

[14] C. Kleiner and A. Koschel, "Towards Automated Generic Electronic Flight Log Book Transfer," 15^{th} IC on Business Information Systems (BIS 2012), Springer, Germany, LNBIP 117, 2012, pp. 177–188.

[15] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," The VLDB Journal, Springer, U.S.A, vol. 10(4), Dec. 2001, pp. 334–350.

[16] A. Boukottaya and C. Vanoirbeek, "Schema matching for transforming structured documents," Proc. 2005 ACM symposium on Document engineering. ACM, U.S.A, pp. 101–110.

[17] A. Morishima, T. Okawara, J. Tanaka, and K. Ishikawa, "SMART: A tool for semantic-driven creation of complex XML mappings," Proc. SIGMOD 2005. ACM, New York, NY, U.S.A., pp. 909–911.

[18] Z. Youzhi, P. Peng, and Z. Geng, "Design of History Database for Networked Control Systems," CCC2007, 2007, pp. 292–296.

[19] L. Checiu and D. Ionescu, "A new algorithm for mapping XML Schema to XML Schema," International Joint Conference on Computational Cybernetics and Technical Informatics (ICCC-CONTI), May 2010, pp. 625–630.

[20] M. Roth, M. Hernandez, P. Coulthard, et. al., "XML mapping technology: Making connections in an XML-centric world," IBM Systems Journal, vol.45, no.2, 2006, pp. 389–409.

A Decomposition-based Method for QoS-aware Web Service Composition with Large-scale Composition Structure

Lianyong Qi[1,2], Xiaona Xia[1], Jiancheng Ni[1],Chunmei Ma[1], Yanxue Luo[1]
[1] Computer Science College
Qufu Normal University, P.R. China
[2] State Key Lab. for Novel Software Technology
Nanjing University, P.R. China
Email: lianyongqi@gmail.com

***Abstract*—Web service composition (WSC), has been considered as a promising way for integrating various distributed computing resources for complex application requirements. However, for a QoS-aware WSC problem with large composition structure, much computation time is necessary to determine the QoS-optimal composite solution, which challenges the service composition applications in large-scale collaboration environment. In view of this challenge, a Decomposition-based service Composition Method, named DCM, is introduced in this paper. Firstly, the proposed DCM method decomposes the large-scale composition structure into many small-scale composition segments, through mixed integer programming. Then for each small-scale composition segment, find a QoS-optimal composite solution with less time cost. Through experiments, we demonstrate that the execution efficiency of DCM outperforms the present service composition methods, especially for the WSC problems with large-scale composition structure.**

Keywords-web service composition; QoS; Decomposition; mixed integer programming

I. INTRODUCTION

In recent years, the web service technology has gained more and more attention and is becoming the de facto standard for integrating various distributed computing resources for complex application requirements [1-3]. Through functional encapsulation, a web service could be advertised by the service provider, and invoked by an end user via the pre-provided accessible interfaces. The easy-to-use property of web service brings more convenience for both service providers and end users, and greatly benefits the flexible integration of cross-domain applications.

However, the function that a single web service could provide is usually limited, compared to the complex computing requirements from end users. Therefore, to compose various component services into a more powerful composite service (WSC) has been considered as a promising way to satisfy the end users' complex requirements. For example, more and more WSC instances are deployed in the popular areas, e.g., Scientific Workflow, Electronic Commerce and Multimedia Delivery applications [4]. However, as there are many web services that share similar functionality, the candidate composite solutions for a WSC problem are multiple, not unique. In this situation, quality of service (QoS) could be recruited as an important discriminating factor, because a WSC process is usually accompanied with various QoS constraints from end users. For example, a smart-phone end user may expect that the total *latency time* of the composite multimedia service not exceed 2 seconds. Therefore, for a WSC problem, the next question is to find a QoS-optimal composite solution from the huge amount of candidates, while considering the various global QoS constraints from end users (QoS-aware web service composition).

Many researchers concentrate on this hot research topic and put forward some valuable methods for the QoS-aware WSC problems [4-9]. However, these proposed methods cannot work very well because of inefficiency. This is due to the fact that QoS-aware WSC is inherently a NP-hard problem [6]' so it is usually time-consuming to find the QoS-optimal composite solution. Especially for the WSC problem with large-scale composition structure (i.e., the WSC process consists of many component tasks. For example, an e-Science composite process with dozens of tasks), the traditional service composition methods may fail in delivering satisfactory results within limited time period. Hence, it is of great challenge to study more efficient composition method. In view of this challenge, a novel Decomposition-based service composition method, named DCM, is proposed. DCM is based on mixed integer programming, which is an optimization approach that has a set of goal function that should be maximized or minimized and a set of constraint conditions that should be satisfied.

The remainder of the paper is organized as follows. In Section Ⅱ, we put forward our motivation via a large-scale service composition example, and introduce the necessary preliminary knowledge. A Decomposition-based service composition method, named DCM, is proposed in Section Ⅲ. Through the experiments in Section Ⅳ, we demonstrate the feasibility and efficiency of DCM in solving the WSC problems with large-scale composition structure. In SectionⅤ, the proposed DCM method is evaluated, and finally we summarize the paper and point out our future research directions in SectionⅥ.

ISBN: 978-1-61208-270-7

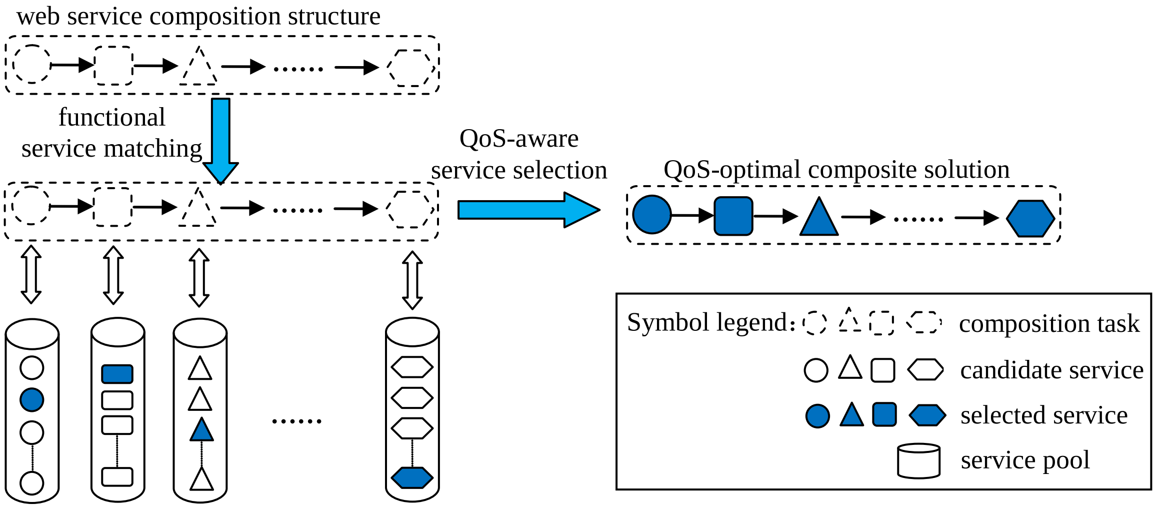


Figure1. An example of QoS-aware web service composition process

II. Preliminary Knowledge and Motivation

In order to facilitate the further discussion, some preliminary knowledge of QoS-aware WSC problem, is introduced here. Concretely, some basic concepts are listed as below.

1. ***TK***={tk_1, …, tk_i, …, tk_n}. tk_i ($1 \leq i \leq n$) denotes a composition task consisted in the service composition structure.

2. ***CR***={cr_1, …, cr_j, …, cr_m}. cr_j ($1 \leq j \leq m$) is a QoS criterion of web service and *m* is the number of QoS constraints requested by an end user. Commonly, for a web service *ws*, its quality value over QoS criterion cr_j could be denoted by $ws.cr_j$.

3. $\boldsymbol{CST_{global}}$={$cst_1$, …, cst_j, …, cst_m}. cst_j ($1 \leq j \leq m$) is a global QoS constraint over criterion $cr_j \in$ ***CR*** by an end-user and *m* is the number of QoS constraints.

4. $\boldsymbol{Pool_i}$={ ws_i^1 , …, ws_i^k , …, ws_i^l } is a service pool corresponding to composition task $tk_i \in$ ***TK***. Namely, each service $ws_i^k \in \boldsymbol{Pool_i}$ ($1 \leq k \leq l$) could execute task tk_i($1 \leq i \leq n$), and *l* is number of functional qualified candidate services for task tk_i.

5. ***WGT*** = {wgt_1, …, wgt_j, …, wgt_m} is the weight value set for different QoS criteria. $wgt_j \in$ ***WGT*** ($1 \leq j \leq m$) is the weight value for QoS criterion cr_j. Generally, set ***WGT*** could be available from the end-user's preferences for different QoS criteria.

6. ***CompS*** = { $ws_1^{k_1}$, …, $ws_i^{k_i}$, …, $ws_n^{k_n}$ } is a functional qualified composite solution, where each component service $ws_i^{k_i}$ is the k_i-th candidate in $\boldsymbol{Pool_i}$ of task tk_i($1 \leq i \leq n$).

For example, as in Fig. 1, assume that there are four tasks in a composition process, then ***TK***={tk_1, tk_2, tk_3, tk_4}. Assume that for each task node, there are four candidate services; Then, for task node tk_i ($1 \leq i \leq 4$), its service pool $\boldsymbol{Pool_i}$ ={ ws_i^1 , …, ws_i^4 }. If we only consider two QoS criteria: *latency* and *reputation*, whose constraints are respectively *latency*<1s and *reputation*>98%, then set ***CR***={*latency*, *reputation*} and $\boldsymbol{CST_{global}}$={*latency* ∈ (0s, 1s), *reputation* ∈ (98%, 100%)}. If ***WGT*** = {wgt_{la}, wgt_{re}} where wgt_{la} = 0.7 and wgt_{re} = 0.3, then it means that *latency* is more important than *reputation* for the end user. Consider the example in Fig. 1, set { ws_1^2 , ws_2^1 , ws_3^3 , ws_4^4 } is a composite solution that belongs to set ***CompS***.

Next, with the introduced basic concepts and the example in Fig. 1, we can clarify our motivation more clearly and formally. As there are *n* composition tasks {tk_1, …, tk_n} in set ***TK***, and for each task tk_i ($1 \leq i \leq n$) in ***TK***, there are *l* functional qualified candidates { ws_i^1 , …, ws_i^l } in service pool $\boldsymbol{Pool_i}$, the total number of functional qualified composite solutions is l^n. Next, our goal is to find a QoS-optimal composite solution ***CompS*** from all the l^n ones, while considering the end user's global QoS constraints $\boldsymbol{CST_{global}}$ and weight value set ***WGT***.

However, for a WSC problem with large-scale composition structure, the number of composition tasks (i.e., *n*) is usually large, which may lead to much time cost in order to find a QoS-optimal composite solution and disappoints the end user. In view of this, we put forward a composition method DCM, in the next section. DCM can decompose the large-scale composition structure into several small-scale ones, through which the composition efficiency could be improved significantly.

TABLE I. AGGREGATION TYPES AND AGGREGATION FUNCTIONS OF QOS CRITERIA

Aggregation type	Criterion	Aggregation function
Summation	price, duration	$CompS.cr_j = \sum_{i=1}^{n} ws_i.cr_j$
Average	reputation	$CompS.cr_j = \frac{1}{n}\sum_{i=1}^{n} ws_i.cr_j$
Multiplication	availability, success rate	$CompS.cr_j = \prod_{i=1}^{n} ws_i.cr_j$

III. A SERVICE COMPOSITION METHOD: DCM

In this section, a service composition method DCM is proposed, to improve the efficiency of WSC problems with large-scale composition structure.

A. *Hypotheses*

For the convenience of further discussions, some hypotheses are declared firstly.

Hypothesis1: Only negative QoS criteria (i.e., the smaller its value is, the better it is for user) are considered, e.g., *composition time*, *composition price*, as positive criteria could be transformed into negative ones by multiplying -1.

Hypothesis2: Only the sequential composition model illustrated in Fig. 1 is discussed, as other composition models (e.g., parallel, alternative and loops) could be transformed into the sequential model by present mature unfolding techniques [4-5].

Hypothesis3: In the sequential composition model, the aggregation types of various QoS criteria are different. In this paper, the common aggregation types and aggregation functions introduced in [4] are employed as in Table 1, where *CompS* denotes a composite solution and *n* is the number of tasks in composition structure.

B. *A QoS-aware service composition Method:* DCM

The main idea of our proposed DCM is: Firstly, the large-scale composition structure with *n* tasks is decomposed into $\lceil n/k \rceil$ small-scale composition segments, each with *k* ($1 \le k \le n$) tasks. Secondly, calculate the segment QoS constraints for each composition segment, through mixed integer programming. Thirdly, for each composition segment, find a QoS-optimal composite solution that satisfies the segment QoS constraints. Next, we will introduce these three steps of proposed DCM method separately.

(1) Step1: Decompose the large-scale composition structure into $\lceil n/k \rceil$ small-scale composition segments.

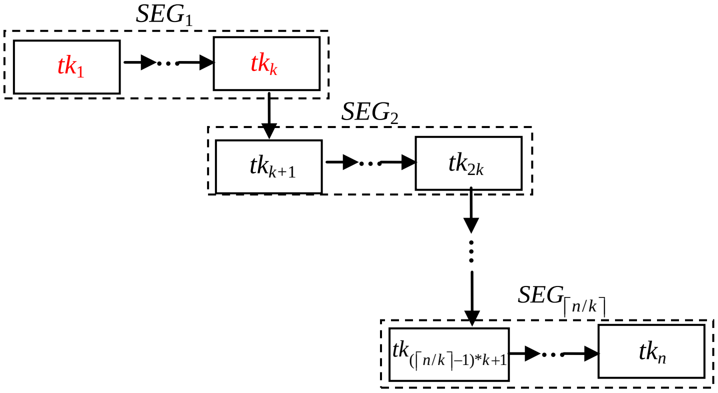


Figure2. Decompose the large-scale composition structure into small-scale composition segments

As the composition structure is large-scale (i.e, *n* is large), the needed time cost for finding a QoS-optimal composite solution may disappoints the end user. Therefore, we firstly decompose the large-scale composition structure into $\lceil n/k \rceil$ small-scale composition segments, each with *k* ($1 \le k \le n$) tasks. For example, as illustrated in Fig. 2, the composition structure with *n* tasks is decomposed into $\lceil n/k \rceil$ composition segments, which are denoted by SEG_1=(tk_1, …, tk_k), SEG_2=(tk_{k+1}, …, tk_{2k}), …, $SEG_{\lceil n/k \rceil}$=($tk_{(\lceil n/k \rceil - 1)*k+1}$, …, tk_n).

(2) Step2: Decompose the global QoS constraints *CST*$_{global}$ into $\lceil n/k \rceil$ segment QoS constraints via mixed integer programming.

After achieving $\lceil n/k \rceil$ small-scale composition segments, in this step, we calculate the segment QoS constraints for each composition segment, by decomposing the global QoS constraints $\boldsymbol{CST_{global}}$ via mixed integer programming technique.

Firstly, by the mathematical statistic technique, we calculate the minimal and maximal values over QoS criterion cr_j ($1 \le j \le m$) of candidate services for each task tk_i ($1 \le i \le n$), which are denoted by min_i^j and max_i^j respectively. Then a value range [min_i^j , max_i^j] is achieved, which depicts the QoS criterion cr_j's value distribution of task tk_i's candidate services. Then, we calculate the value distribution of candidates of each composition segment SEG_p ($1 \le p \le \lceil n/k \rceil$) over QoS criterion cr_j, which is denoted by [MIN_p^j , MAX_p^j], according to the aggregation functions introduced in TABLE Ⅰ. Afterwards, range [MIN_p^j , MAX_p^j] is discretized into *d* ($d \ge 2$) discrete value with interval dis_p^j =(MAX_p^j - MIN_p^j)/(*d*-1), i.e., { ${}^{1}Bound_p^j$,…, ${}^{d}Bound_p^j$ }, where ${}^{1}Bound_p^j$ = MIN_p^j and ${}^{d}Bound_p^j$ = MAX_p^j . Here, each discrete value ${}^{x}Bound_p^j$ ($1 \le x \le d$) corresponds to the composition segment SEG_p's QoS constraint [0, ${}^{x}Bound_p^j$] over QoS criterion *cr*cr_j. Our final object of this step is to find an appropriate x value for each QoS criterion cr_j ($1\le j\le m$) of each segment SEG_p ($1\le p\le\lceil n/k\rceil$).

Next, we utilize the mixed integer programming to determine the best segment constraint [0, ${}^{x}Bound_p^j$] for each QoS criterion cr_j of each composition segment SEG_p. According to the mixed integer programming, this optimization problem could be formalized into the following question:

$$\text{Maximize } \prod_{p=1}^{\lceil n/k\rceil}\prod_{j=1}^{m} Utility({}^{x}Bound_p^j) \tag{1}$$

$$\text{Subject to } \prod_{p=1}^{\lceil n/k\rceil} {}^{x}Bound_p^j \le cst_j \tag{2}$$

$$\frac{1}{\lceil n/k\rceil}\sum_{p=1}^{\lceil n/k\rceil} {}^{x}Bound_p^j \le cst_j \tag{3}$$

$$\sum_{p=1}^{\lceil n/k\rceil} {}^{x}Bound_p^j \le cst_j \tag{4}$$

$$x\in\{1, 2, \dots, d\} \tag{5}$$

Here, $Utility({}^{x}Bound_p^j)$ in (1) denotes the "goodness" of utilizing [0, ${}^{x}Bound_p^j$] as segment SEG_p's QoS constraint over criterion cr_j, whose computation manner could be found in [5], so we will not discuss it here. The left parts of (2)-(4) respectively denote the aggregated segment QoS constraints, corresponding to the three aggregation types in TABLE Ⅰ, which should not exceed the end user's global QoS constraint. The constraint condition in (5) means that variable x has d possible values. After solving the above optimization problem with mixed integer programming, we can derive the best segment QoS constraints [0, ${}^{x}Bound_p^j$]($1\le j\le m$) for each composition segment SEG_p ($1\le p\le\lceil n/k\rceil$).

(3) Step3: For each composition segment, find the QoS-optimal composite solution that satisfies segment QoS constraints.

In Step1, we obtain $\lceil n/k\rceil$ small-scale composition segments { SEG_1, SEG_2,…, $SEG_{\lceil n/k\rceil}$ }, each with k ($1\le k\le n$) tasks. Through which, we successfully transform a large-scale WSC problem into $\lceil n/k\rceil$ small-scale WSC sub-problems. And in Step2, for each WSC sub-problem, its QoS constraints are also obtained. So in Step3, we only need to solve these QoS constrained WSC sub-problems. Next, we formalize the WSC sub-problem (corresponding to a composition segment) as follows:

$$\text{Maximize } Utility(SEG_p) \tag{6}$$

Subject to

$$\prod_{q=(p-1)*k+1}^{p*k}\left(\sum_{z=1}^{l} ws_q^z.cr_j * x_q^z\right) \le {}^{x}Bound_p^j \tag{7}$$

$$\frac{1}{\lceil n/k\rceil}\left(\sum_{q=(p-1)*k+1}^{p*k}\sum_{z=1}^{l} ws_q^z.cr_j * x_q^z\right) \le {}^{x}Bound_p^j \tag{8}$$

$$\sum_{q=(p-1)*k+1}^{p*k}\sum_{z=1}^{l} ws_q^z.cr_j * x_q^z \le {}^{x}Bound_p^j \tag{9}$$

$$\sum_{z=1}^{l} x_q^z = 1\, x\in\{1, 2, \dots, d\} \tag{10}$$

$$x_q^z \in\{0, 1\} \tag{11}$$

Here, $Utility(SEG_p)$ in (6) denotes the utility value of composite solution for SEG_p. The left parts of (7)-(9) respectively denote the aggregated value for composite solution of segment SEG_p, over QoS criterion cr_j, which should not exceed the calculated segment QoS constraints, according to the three aggregation types in TABLE Ⅰ. The constraint conditions in (10)-(11) mean that for a segment task, only one candidate service would be selected, to form a composite solution. After solving the above mixed integer programming problem, a QoS-optimal composite solution *CompS-p* could be achieved for composition segment SEG_p. And finally, the set {*CompS*-1, …, *CompS*-$\lceil n/k\rceil$ } makes up the QoS-optimal composite solution, for the original WSC problem with large-scale composition structure.

IV. Experiment

In this section, a group of experiments are enacted and executed to validate the effectiveness and efficiency of proposed DCM method, especially in dealing with the WSC problems with large composition structure.

A. Experiment Configuration

In this paper, the experiment is based on the updated *QWS Dataset* from Dr. Eyhab Al-Masri [5], which is widely used in the service computing domain. The experiments were deployed and executed on a Lenovo machine with Intel Pentium 2.40 GHz processors and 1.00 GB RAM. The machine is running under Windows XP and C++ integration environment.

B. Experiment Results and Analyses

In order to evaluate the feasibility and efficiency of DCM in dealing with WSC problems with large-scale composition structure, the following four test profiles are

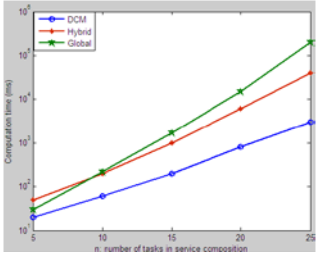


Figure3. Performance comparison w.r.t. number of tasks

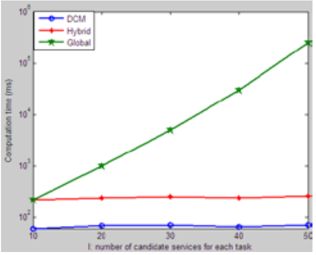


Figure4. Performance comparison w.r.t. number of candidates

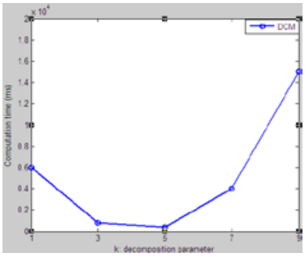


Figure5. Performance comparison w.r.t. parameter *k*

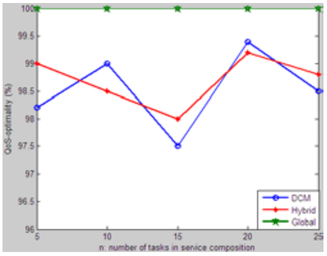


Figure6. QoS optimality w.r.t. number of tasks

designed and compared with another two present methods: *Global* [6] and *Hybrid* [5]. In the experiments, the parameter *d* is equal to 5 for both *Hybrid* and DCM, and five QoS constraints are employed for all test cases.

***Profile*1: performance comparison of** DCM**, *Global* and *Hybrid* with different task number (i.e., *n*)**

In this profile, the computation time of three methods are tested and compared, for finding a QoS-optimal composite solution. In the experiment setting, the number of tasks in composition structure (i.e., *n*) is varied from 5 to 25. And for each task, there are 10 candidate services, i.e., *l*=10. The parameter *k* in DCM is equal to 3. The experiment comparison results are demonstrated in Fig. 3. As can be seen from Fig. 3, the computation time of DCM outperforms the other two methods when *n* is increased. Especially when the value of *n* is large, the proposed DCM method exhibits its significant advantages, in dealing with WSC problems.

***Profile*2: performance comparison of** DCM**, *Global* and *Hybrid* with different candidates (i.e., *l*)**

In this profile, the computation time of three methods is compared when the number of candidate services changes. In the experiment setting, the number of tasks in composition structure, i.e., *n*=10. And for each task, the number of candidate services (i.e., *l*), is varied from 10 to 50. The parameter *k* in DCM is equal to 3. The experiment comparison results are demonstrated in Fig. 4. As in Fig. 4, the computation time of *Global* increases significantly when *l* arises. However, DCM and *Hybrid* exhibit similar change trends, whose computation time both stay nearly unchanged with the increase of *l*.

Profile*3: performance comparison of** DCM **with different values of parameter *k

In our proposed DCM method, we improve the composition efficiency, by transforming a large-scale WSC problem with *n* tasks into $\lceil n/k \rceil$ small-scale WSC sub-problems. Therefore, *k* is an important parameter in DCM. In this profile, we observe the performance change of DCM with the increase of *k*. In the experiment setting, the number of tasks in composition structure, i.e., *n*=20. And for each task, the number of candidate services, i.e., *l* is equal to 10, while the value of parameter *k* is varied from 1 to 9, with interval of 2. The experiment comparison results are demonstrated in Fig. 5. As can be seen from Fig. 5, the computation time of DCM decreases firstly with the arise of the *k*, this is because when *k* grows, the computation workload for achieving segment QoS constraints reduces. While when *k* grows further, the computation workload for finding segment QoS-optimal composite solution increase, so the computation time arises correspondingly.

Profile*4: QoS optimality comparison of** DCM**, *Global* and *Hybrid

Execution time is an important evaluation item for QoS-aware web service composition, however, the QoS optimality of obtained composite solution should also be considered. In this profile, we test the QoS optimality of three methods. In the experiment setting, the number of candidate services for each task, i.e., *l* is fixed to 10, *n* is varied from 5 to 25, the employed parameter *k* (the task length of each composition segment) is equal to $\lceil n/3 \rceil$. The experiment results are shown in Fig. 6. As in Fig. 6, the *Global* method is the best in terms of QoS optimality of obtained composite solution. And the QoS optimality of DCM is almost the same as that of *Hybrid*, both close to 100%. Therefore, DCM can generate a QoS near-to-optimal composite solution, while ensures better composition efficiency.

V. EVALUATION

In this section, we will analyze the time complexity of DCM. A comparison with related work is also presented, to show the advantages of our proposed DCM method.

A. *Complexity Analysis*

In this subsection, the time complexity of the three steps of DCM will be analyzed separately, after which we obtain the total time complexity of DCM.

(1)Step1: Decomposition of large-scale composition structure into small-scale ones

With the determined *k* value in DCM, we can decompose the large-scale composition structure with n tasks into $\lceil n/k \rceil$ small-scale composition segments, whose time complexity is O(1).

(2)Step2: Determine the segment QoS constraints for all composition segments

A MIP problem is achieved, for determining the segment QoS constraints for each composition segment. Therefore, its time complexity is O($2^{\lceil n/k \rceil * m * d}$), where n is the task number, k is the decomposition parameter employed in DCM, m is the number of QoS constraints, and d is the discretization parameter.

(3)Step3: Determine the QoS-optimal composite solution

The QoS-optimal sub-composition solutions of $\lceil n / k \rceil$ composition segments make up the QoS-optimal composite solution, for the original WSC problem with large-scale composition structure. For each composition segment, the time complexity is O(2^{k*m*d}), for determining the QoS-optimal sub-composition solution. Therefore, the time complexity of Step3 is O($\lceil n / k \rceil$ *2^{k*m*d}).

According to the above analysis, a conclusion could be made that the time complexity of our proposed DCM is O($2^{\lceil n/k \rceil * m * d}$ + $\lceil n / k \rceil$ *2^{k*m*d}) = O($2^{\lceil n/k \rceil * m * d}$), as k is a fixed value in DCM. Therefore, DCM outperforms *Global* and *Hybrid* methods, whose time complexity are O(2^{n*l}) and O(2^{n*m*d}) respectively, when dealing with WSC problems with large-scale composition structure.

B. Related Work and Comparison Analysis

QoS-aware web service composition has been a hot research topic in the domain [1-3]. In [6], a *Global* method is introduced, where the QoS-aware WSC problem is modeled into a mixed integer programming one, whose time complexity is still large. To reduce the time complexity of WSC, many works focus on minimizing the number of candidate services for each task, such as the *Hybrid* method [5], the *LOEM* method [4], *Skyline* method [7], *MOCACO* method [8] and heuristic method [9]. However, their time complexity is still exponential with the task number, which may lead to much time cost when the composition task number is large. In this paper, we transform the large-scale WSC problem into several small-scale ones, and put forward a decomposition-based composition method DCM. Through experiments, we demonstrate the feasibility and efficiency of DCM in dealing with the WSC problems with large-scale composition structure.

VI. Conclusions

In this paper, a decomposition-based method, named DCM, has been presented for the WSC problems with large-scale composition structure. Through experiments, we demonstrate the feasibility of DCM. However, there are some shortcomings in the paper; for example, the employed parameter k is not assigned a concrete value, which will be studied as a future research topic.

Acknowledgment

This paper is supported by the Open Project of State Key Lab. for Novel Software Technology (No. KFKT2012B31), Natural Science Foundation of Shandong Province of China (No.ZR2012FQ011), SRI of SPED (No.J12LN06), DRF and UF (BSQD20110123, XJ201227) of QFNU.

References

[1] Yuhong Yan, Min Chen, and Yubin Yang, “Anytime QoS optimization over the PlanGraph for web service composition,” Proceedings of ACM Symposium on Applied Computing (SAC 12), ACM Press, Mar. 2012, pp. 1968-1975, doi:10.1145/2245276.2232101.

[2] Marisol Garcia Valls, R. Fernández-Castro, Iria Estevez Ayres, Pablo Basanta Val, and Iago Rodriguez, “A Bounded-time Service Composition Algorithm for Distributed Real-time Systems,” Proceedings of IEEE International Conference on High Performance Computing and Communication (HPCC 12), IEEE Press, Jun. 2012, pp. 1413-1420, doi:10.1109/HPCC.2012.207.

[3] Wada Hiroshi, Suzuki Junichi, Yamano Yuji, and Oba Katsuya, “E3: A Multiobjective Optimization Framework for SLA-Aware Service Composition,” IEEE Transactions on Services Computing, vol. 5, Sep. 2012, pp. 358-372, doi: 10.1109/TSC.2011.6.

[4] Lianyong Qi, Ying Tang, Wanchun Dou, and Jinjun Chen, “Combining Local Optimization and Enumeration for QoS-Aware Web Service Composition,” Proceedings of IEEE International Conference on Web Services (ICWS 10), IEEE Press, Jul. 2010, pp. 34-41, doi: 10.1109/ICWS.2010.62.

[5] Mohammad Alrifai and Thomass Risse, “Combining global optimization with local selection for efficient QoS-aware service composition,” Proceedings of 18th International Conference on World WideWeb (WWW 09), ACM Press, Apr. 2009, pp. 881-890, doi: 10.1145/1526709.1526828.

[6] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang, “QoS-aware middleware for web services composition,” IEEE Transactions on Software Engineering, May 2004, vol. 30, pp. 311-327, doi: 10.1109/TSE.2004.11.

[7] Mohammad Alrifai, Dimitrios Skoutas, and Thomas Risse. “Selecting skyline services for QoS-based web service composition,” Proceedings of 19th International Conference on World Wide Web (WWW 10), ACM Press, Apr. 2010, pp. 11-20, doi: 10.1145/1772690.1772693.

[8] Wang Li and He Yan-xiang, “A Web Service Composition Algorithm Based on Global QoS Optimizing with MOCACO,” Lecture Notes in Computer Science, vol.6082, Nov. 2011, pp. 79–86, doi:10.1007/978-3-642-13136-3_22.

[9] Adrian Klein, Fuyuki Ishikawa, and Shinichi Honiden, “Efficient Heuristic Approach with Improved Time Complexity for QoS-Aware Service Composition,” Proceedings of IEEE International Conference on Web Services (ICWS 11), IEEE Press, Jul. 2011, pp. 436-443, doi: 10.1109/ICWS.2011.60