MOPAS 2013

The Fourth International Conference on Models and Ontology-based Design of Protocols, Architectures and Services

ISBN: 978-1-61208-266-0

April 21 - 26, 2013

Venice, Italy

MOPAS 2013 Editors

Andreas Schmidt, Karlsruhe University of Applied Sciences & Karlsruhe Institute of Technology, Germany

MOPAS 2013

Foreword

The Fourth International Conference on Models and Ontology-based Design of Protocols, Architectures and Services (MOPAS 2013), held between April 21^{st}-26^{th}, 2013 in Venice, Italy, proposed a new context for presenting achievements, surveys and perspectives in the areas of design, architecture and implementation based on ontologies and related models.

We take here the opportunity to warmly thank all the members of the MOPAS 2013 Technical Program Committee. The creation of such a 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 MOPAS 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 MOPAS 2013 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that MOPAS 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 models and ontology-based design and protocols, architectures and services.

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

MOPAS Advisory Committee:

Raj Jain, Washington University in St. Louis, USA
Marc Lacoste, Orange Labs, France

MOPAS 2013

Committee

MOPAS Advisory Committee

Raj Jain, Washington University in St. Louis, USA
Marc Lacoste, Orange Labs, France

MOPAS 2013 Technical Program Committee

Mourad Alia, Orange Business Services- Grenoble, France
Shawn Bower, Gonzaga University, USA
João Caldeira, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
John R. Callahan, ONR Global London, UK
Shu-Ching Chen, Florida International University - Miami, USA
Vanea Chiprianov, Telecom Bretagne, France
Dickson Chiu, Dickson Computer Systems, Hong Kong
Michel Diaz, LAAS, France
Rachida Dssouli, Concordia University, Canada
Antti Evesti, VTT Technical Research Centre of Finland, Finland
Marcello Frixione, DAFIS - University of Genova, Italy
Bin Guo, Northwestern Polytechnical University, China
Robert C. H. Hsu, Chung Hua University, Taiwan
Sergio Ilari, University of Zaragoza, Spain
Zahid Iqbal, University Graduate Center (UNIK) - Kjeller, Norway
Brigitte Jaumard, Concordia University - Montreal, Canada
Manfred A. Jeusfeld, Tilburg University, The Netherlands
Achilles Kameas, Hellenic Open University, Greece
Marc Lacoste, Orange Labs, France
Thomas D. Lagkas, University of Western Macedonia, Greece
Gianluca Liguori, Bluedeep, Italy
Pericles Loucopoulos, Harokopio University of Athens, Greece
Michele Melchiori, Università degli Studi di Brescia, Italy
Vlad Nicolicin Georgescu, Université de Nantes / SP2 Solutions - La Roche sur Yon, France
Peera Pacharintanakul, TOT, Thailand
Arun Prakash, Fraunhofer Institute for Open Communication Systems (FOKUS) - Berlin, Germany
Neeli R. Prasad, Aalborg University, Denmark
Gustavo Rossi, Universidad Nacional de La Plata, Argentina
Wieland Schwinger, Johannes Kepler University Linz, Austria
Vasco Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Shensheng Tang, Missouri Western State University, USA
Zoltan Theisz, evopro Informatics and Automation, Ltd., Hungary

Dimitrios D. Vergados, University of Piraeus, Greece
Roberto Willrich, Santa Catarina Federal University, Brazil
Nataša Živic, University of Siegen, Germany

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Design Pattern for Pervasive Systems

Fatma Achour, Anis Jedidi, Gargouri Faiez
MIRACL
Multimedia, Information Systems and Advanced Computing Laboratory
Sfax University, Tunisia
E-mails: {fatma.achour@gmail.com, anis.jedidi@isimsf.rnu.tn, faiez.gargouri@isimsf.rnu.tn}

Abstract-**The ubiquitous computing, also named pervasive computing, gathers the characteristics of mobile computing and the techniques of context-awareness which are actually flexible, adaptable, and capable of acting autonomously. However, pervasive computing introduces a variety of software, hardware and users' engineering challenges. We look, while realizing our thesis, for a framework for the design and adaptation of the pervasive system. Firstly, we attempt to allow the software agents to assist the users in manipulating the resources on the Web. Then, we try to manage the interoperability between the resources and their adaptation to the various contexts of use (semantic, network, etc.). Finally, we present the knowledge in an accessible form to both man and machine. In this paper, we present the first phase of our work which resides in conceiving a structure for the design pattern of the pervasive system.**

Keywords - pervasive computing; models; transformation; MDA.

I. Introduction

The pervasive system combines ubiquity, mobility and context-awareness. The evolution of pervasive computing, fortunately, has opened new horizons to the classical information systems by integrating new technologies and services that may lead to a seamless access to information sources at anytime, anyhow and anywhere. But, for every plus there is a minus. This evolution has brought about new challenges to information modeling. No matter what the challenge is, these systems must be able to be used in various contexts according to the user environment, his profile, the used terminal, etc.

In the literature, several researchers are working on the pervasive systems adaptation. Each researcher offers his own dimensions for the context. In this paper, we intend to realize a design pattern structure useful for the adaptation of any application in a pervasive context. According to Alexander [1], a pattern describes a problem which frequently occurs in an environment as well as a solution that can be adapted to a specific situation.

This paper is organized as follows. Section 2 presents the creation steps of the Pattern Design. Section 3 presents the process to validate the design pattern structure.

II. State of the Art

Daniel Salber defines the context in [2] as the "Environmental information or context that covers information that is a part of an application operating environment and that can be sensed by the application. This typically includes the location, identity, activity and the state of the people, groups and objects. The context may also be related to places or the computing environment". For such a literature keen on the context model, Held - the oldest model - is deemed to be the simplest key value model to use [2]. After this model, more and more typical models came into sight. The models are often classified by the structure plans of data which are used to describe and transmit the contextual information. In this work, we present four models: CSCP, SOUPA, COMANTO and Activity.

A. CSCP (Comprehensive Structured Context Profiles)

In 2002, CSCP, which is the extension of CC/PP (Composite Capabilities / Preferences Profiles), was proposed by Held who did not define any fixed hierarchy to solve the problems of CC/PP. The context model CSCP represents the profile sessions and is based on RDF; it does not impose any fixed hierarchical structure for the context notion. Thus, it inherits the full flexibility and the expressive power of RDF. The CSCP, indeed, allows the merger of the profile fragments that are dynamically retrieved even from various web sites [3].

- The user profile: is composed of static characteristics (name, first name, etc.) and evolutionary characteristics which are defined by his environment (location, time, etc.) and his preferences
- The device profile: presents the material context (type of device, screen size, etc.) and the software context (operating system, version, etc.).
- The network profile: exposes information about the type of network, its characteristics, etc.
- The session profile: presents the connection from the user to the system.

B. SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications)

The SOUPA project was created in November, 2003. It is part of the work caring for the semantic web which was developed by the "Special Interest Unicomp" group [4]. The objective of this project is to define the ontology written in OWL which supports the pervasive applications. The SOUPA concepts cover themselves with the set of existing ontology vocabularies.

The ontology cited by SOUPA included the Friend Of Friend (FOF), DAML Time, Open Cyc, RCC, COBRA-ONT, MoGATU BDI and Rei ontology policy. In other words, SOUPA contains two distinctive connected subsets of ontology: the SOUPA Core and the SOUPA Extension. On the one hand, the first ontology allows defining the

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-266-0

generic vocabularies for the construction of the pervasive applications [5]. On the other hand, the second one allows defining the additional vocabularies to support a specific type of applications and provides examples of defining new extensions of SOUPA.

In this work, we focus on the SOUPA Core. The SOUPA core ontology includes a set of sub-ontology.

- Person: this ontology defines the vocabulary to describe the typical personal information and profile of a person.
- Policy and Action: the security and privacy are two increasing concerns in the development and deployment of the pervasive computing systems.
- Agent and BDI: during the construction of the pervasive systems, it is useful to present the computer model entities as agents.
- Time: SOUPA defined a set of ontology to model the time and the temporal relations.
- Space: this ontology is designed to support reasoning about the spatial relationships between the different types of geographical spaces and geographical coordinates.
- Event: is the event of the activities that have the same spatial and temporal extensions.

C. COMANTO (COntext Management oNTOlogy)

The ontology of context Daidalos, also called COMANTO (COntext management oNTOlogy), was created by Roussaki [6] in 2005. It is the outcome of a hybrid context modeling approach to handle the context objects and the context knowledge. The COMANTO ontology is proposed as a public context semantic vocabulary supporting the efficient reasoning about the contextual concepts (such as users, activities, tools, etc.) and their associations. The ontology classes of the context COMANTO can be described as:

- Person: it is the central entity in the ontology COMANTO. It represents all the human entities and offers the diverse properties of the data type to integrate the user into the related context.
- Place: it is the abstraction of a physical space. It offers a set of data properties that associate a physical location with its symbolic or geographical representation.
- Preferences: in order to model the user, the service, the network and the device preferences.
- Service: it stores the information relevant to the applications taken by the user.
- Legal Entity: This class is mentioned as a representation of the corporate actors involved in the pervasive computing supply chain.
- Device: the class "Device" is an abstract representation of mobile devices.
- Network: the class "Network" contains all the network information.
- Sensors: is another abstract class to get a true representation of the device.

D. ACTIVITY Model

The theory of activity is a descriptive tool to help understand the unity of knowledge and activity [7]. It focuses on the practice of the individual and collective work. An activity consists of a subject, an object and an artifact or a tool of mediation. The activity theory is applied to provide a model that covers all the possible contexts in ubiquitous computing. The model of context "Activity" appears in eight main categories. In each category, there are other levels of context. The elements of the context model can be described as follows:

- User: information about the user that the system is interested in.
- Tools: list of the available tools in the public place.
- Rules: norms, social rules and legislations through which the user is related to the others in his community.
- Community: information about the persons around the user who can have an influence on the activity.
- Object: the intention and the objective of the user.
- Time: it is a moment in a particular situation in which an activity takes place.
- The division of labor: the distribution of tasks between the members of the community.
- Result: the result of the object transformation.

III. DESIGN PATTERN STRUCTURE MODELING PROCESS

The various phases of the design pattern structure conception are presented in the Figure below (see Figure 1).

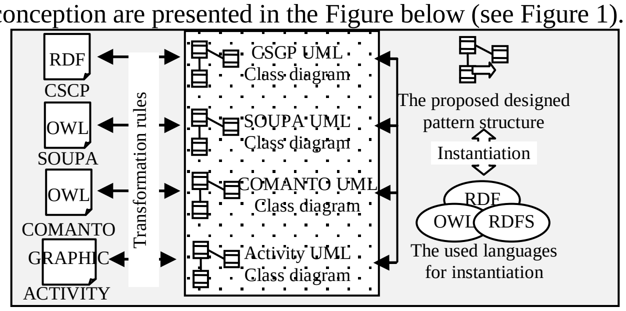


Figure 1. The various steps of the proposal execution.

Firstly, we intend to transform the four contextual models: CSCP, SOUPA, COMANTO and Activity into a UML class diagram. Secondly, we perform a comparative study of the different concepts of each model. Finally, we present a consistent class diagram of these models. These models will be validated by the instances creation in the medical, educational, commercial and archaeological fields.

A. The transformation of contextual models

1) The CSCP model transformation: The CSCP transformation of the graphical RDF (Resource Description Framework) model into a valid class diagram is realized according to the following transformation rules .

The "predicate" and "object" are transformed into UML classes, The "subject" is transformed into association,

The “rdf:bag” is transformed into a composition with a constraint OCL.

The UML class diagram of the CSCP model is shown in the following Figure (see Figure. 2).

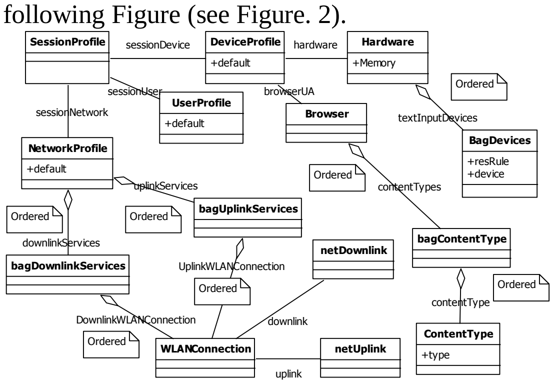


Figure 2. The class diagram of the CSCP model.

2) The SOUPA ontology transformation: The SOUPA core class diagram is achieved through a theoretical study of the different concepts in each ontology. In fact, a set of transformation rules are defined in a manual way to enssure this transformation. The class diagram corresponding to SOUPA ontology is represented in the Figure 3. The transformation rules [8] that allow for the realization of the class diagram represented in Figure 3 are presented as follows:

Owl: Property: is transformed into a UML attribute.
Owl: class: is transformed into a UML class.
Defined by: like “owl: property”, the property “defined by” is transformed into a UML attribute.
Union and intersection: these relations are replaced by the relations of the composition within the OCL constraints.
subClassOf: the property “subClassOf” is transformed into an extended relationship.

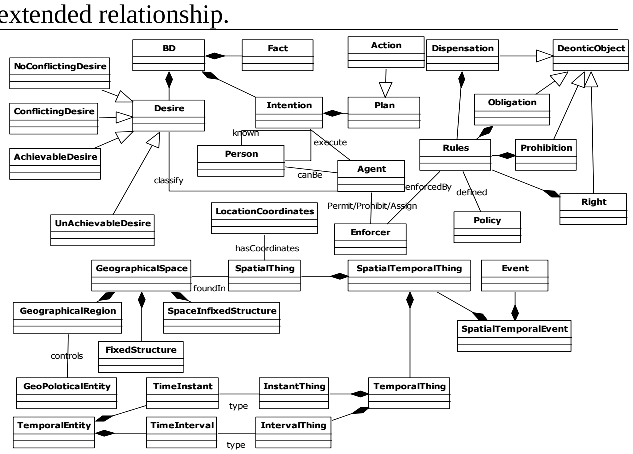


Figure 3. The class diagram of SOUPA Ontology.

We define the following set of OCL constraints from SOUPA Core.

SpatialTemporalEvent=intersect(SpatialTemporalThing, Event)
SpatialTemporalThing=intersect(TemporalThing,SpatialThing)
TemporalThing= Union (InstantThing, IntervalThing)
TemporalEntity= Union (TimeInstant, TimeInterval)
GeographicalSpace= Union(GeographicalRegion, FixedStructure, SpaceInFixedStructure)

3) The COMANTO ontology transformation: The transformation of ontology COMANTO into a corresponding class diagram (see Figure. 4) is carried out according to the following transformation rules [6].

The OWL nodes are transformed into UMLclasses.
Two-way links titled (in reverse) are transformed into a two-way association with two titles.
The unidirectional arcs between the nodes are transformed into a unidirectional association.
The relation “rdfs:subClassOf” is transformed into a heritage relation.
The relation “owl:disjointWith” is transformed into an association with the OCL constraint.

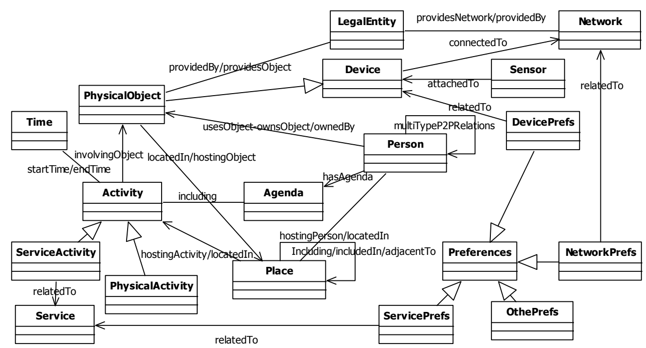


Figure 4. The class diagram of the ontology COMANTO.

4) The ACTIVITY model transformation: The transformation process of the context “Activity” structure into a UML class diagram represents the transformation of a simple graphic model into a UML class Diagram.

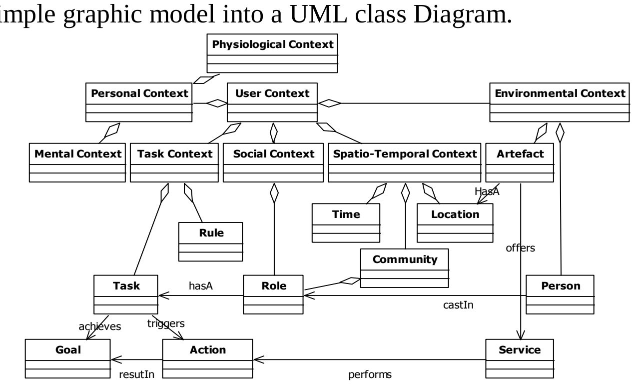


Figure 5. The class diagram of the Activity model.

The result of the Activity model transformation represented in the Figure above (see Figure. 5) is fulfilled through the use of the following three rules.

The nodes are transformed into classes,
The arcs are converted into oriented associations in the UML diagram,
The arcs of the type “partOf” are transformed by the UML

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-266-0

aggregations.

B. The comparative study

The adaptation of the applications to the pervasive system is essentially based on a given model. In this work, we put the stress on the design pattern structure of the adaptation. To unify the existing models, we try to make a structure according to the four suggested models; CSCP, SOUPA, COMANTO and Activity model. This design pattern structure is a UML class diagram that will be turned into an application ontology written in OWL. This ontology will allow for the support system creation to design the pervasive application.

After the theoretical study of the four presented models, we have made a comparison between the different concepts used in each model. We presented in the following table the intersection concepts of selected models (see Table I). The goal is to make an application structure compiled with all of these models. All these concepts will be transformed into a UML class diagram for the design pattern structure. The latter will be used in the phase of adaptation.

TABLE I. THE COMPARATIVE TABLE OF CONCEPTS.

M / C	CSCP	ACTIVITY	COMMANTO	SOUPA
Device	Artifact	Device	Device profile	Agent
Person	Person	Person	User profile	Person
Social	Role (social multi Type)	–	Property : context)	P2P relations known (person)
Rules	Rule	Legal entity	resRule	Rules
Time	Time	Time	–	Time
Place	Location	Place	–	Geographical space
Activity	Task	Activity	–	–
Action	Action	–	–	Action
Network	–	Network	Network profile	–
Desire	–	Preferences	–	BDI:desires
Service	Service	Service	–	–
Location Coordinate	–	Including Included in Adjacent to	–	Location Coordinate

C. The proposed class diagram

After the comparative study between the four models, we can extract the principal classes that can be used in the proposed design pattern structure. Indeed, a class is called principal if it is common among the concepts of the four presented models.

The associations between the proposed classes are presented according to this theoretical study. The result of this study is presented in the next points.

- Agent: this class is used for the presentation of the different actors in a pervasive system. It assembles the human actor and the device actor.
- Persons: present the human actors.
- Device: presents the peripheral devices in a pervasive system.
- Service: presents the services offered by each device.
- Networks: collect the characteristics of the different types of network.
- Location coordinates: present the spatial relation between the different locations in a pervasive system.
- Preferences: present the information profile of the person who realizes the activity in a pervasive system.
- Activity: presents the characteristics of the activity requested by the user.
- Rules: gather the different rules of activity, person and network interacting in a pervasive system.
- Time: presents the characteristics of the temporal and the relation ones of the different things in a pervasive system.
- Locations: represent the characteristics of localizations of the human and mobile devices in a pervasive system.

These classes will be presented in a UML class diagram (see Figure. 6). In a pervasive system, the context is defined by a set of characteristics that are related either to the user (classes; person, preference and role), to the physical environment (classes; location and location coordination), to the network (classes; network and rules), to the activity (classes activity, time and rules), to the devices (class device) or to the service (class services).

A characteristic is represented by a key/value. The attribute specifies the name of the characteristic and its value is given by the same name. For example, a network may be the characteristic of the flow rate. This instance may have the attribute "flow rate" and the value "1 Mbps". The definition of the representation of the PSI needs to be dynamic and scalable since it is composed of one or more characteristics. These characteristics are not defined in advance, but they are defined by the system designer and according to his needs.

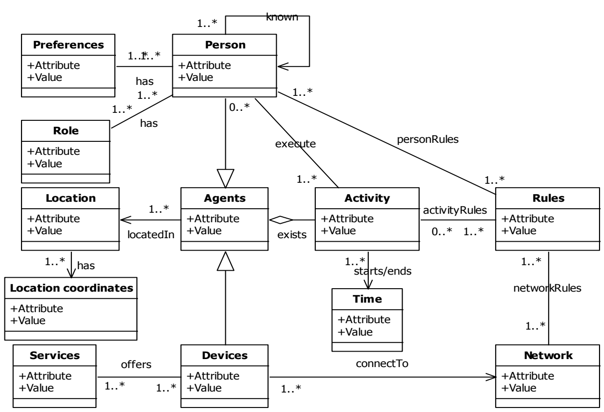


Figure 6. The proposed class diagram.

IV. THE DESIGN PATTERN STRUCTURE VALIDATION

UML is a semi-formal language that allows us to have a better visual clarity of our design pattern structure. However, to describe the instances of this design pattern structure, we have chosen the formalisms RDF /RDFS /OWL. The formal languages are inspired by the logic of description. They are more appropriate to describe the semantics because they provide us with the basic predicates that we can re-use.

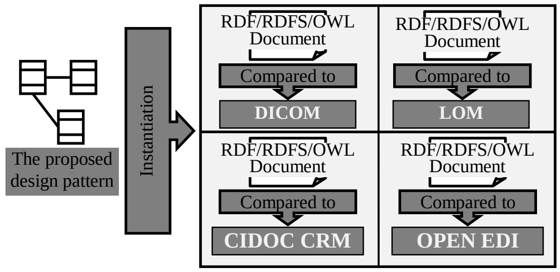


Figure 7. The various steps of validation.

To validate the proposed design pattern structure, we try to apply it to different fields. These instances are created manually and compared with the existing models. We are interested in the validation process of the respective domain ontology of medical, educational, archaeological and commercial field (see Figure 7).

A. The medical field instance

In order to validate the proposed model, we have created an ontology representing an instance of the design pattern structure in the medical field. The classes representing "the mobility" are not represented. The instance created in the medical field will be presented in the next RDF/RDFS/OWL document (see Figure 8). We have compared this instance to the class diagram corresponding to Digital Imaging C0mmunication in Medicine (DICOM).

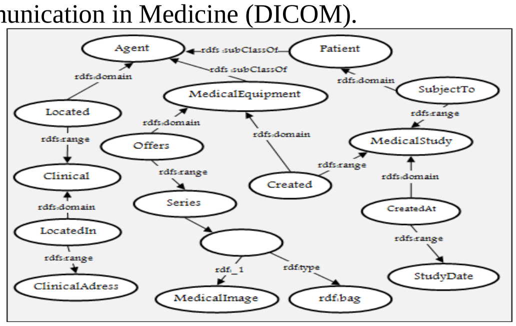


Figure 8. The instance of medical field.

DICOM is a format that defines the methods of connection, transfer and identification of medical data. We found that this instance represents an essential part of this format [9].

After the instance creation, we have created a comparative list of the set of the DICOM classes and the set of instance classes (see Table II). An important number of DICOM concepts are presented in the proposed RDF/RDFS/OWL instance.

TABLE II. THE MEDICAL FIELD COMPARISON.

DICOM	Image	Medical examination	Patient	Medical study	Medical series	Equipment	Terms of Reference	Result	Amendment	Report	Notification content of the study
Instance	*	*	*	*		*	*				

B. The commercial field instance

In order to validate the proposed model, we have created an ontology representing an instance of the design pattern structure in the commercial field. We used the e-commerce open EDI to validate the proposed model.

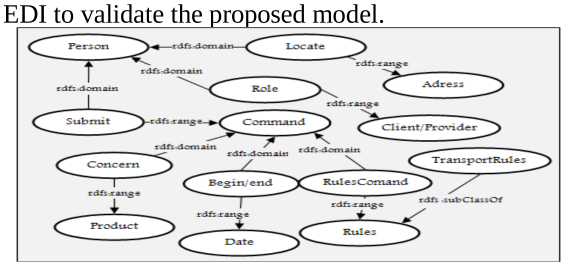


Figure 9. The instance of commercial field.

The open EDI proposes a standard to describe the roles of diverse companies and guarantees that their applications are able to communicate through the EDI messages [10]. To do it, we realized an instance in the commercial field; this instance is presented in Figure 9.

The proposed instance represents a part of the commercial domain. Information in the commercial field has been represented in this instance. The result of the comparison in the commercial field is represented in the following table (see Table III).

TABLE III. THE COMMERCIAL FIELD COMPARISON.

Open EDI	Buyer	Customer contact	Further address	Client	Contact supplier	Supplier	Conditions of Carriage	Header control	Storage location	Locus of control	Currency	Delivery date	Amount including tax	Command post	Place of collection	Place of delivery	Total
Instance	*	*	*	*	*	*	*		*	*		*					

C. The educational field instance

The second field is education. At this level, the standard used for comparison is the Learning Object Meta-data (LOM). The Learning Object Metadata is an international data model, usually encoded in XML, providing a model to describe the meta-data associated with the learning objects of any kind, digital or not [11].

A theoretical study of this standard allowed us to realize the instance of the following model. The instance of this field is represented in the following Figure.

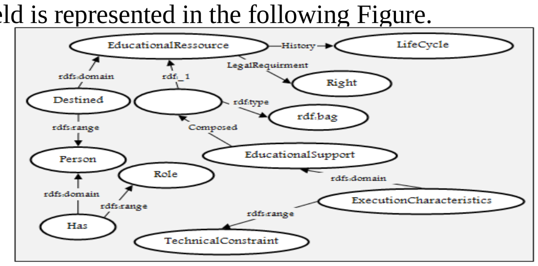


Figure 10. The instance of the educational field.

To compare this instance to the Learning Object Meta-data, we have created a comparative table among the set of

LOM classes and the set of instances. This list is represented in the following table (see Table IV). We found that we have talked about five classes defined in LOM.

TABLE IV. THE EDUCATIONAL FIELD COMPARISON.

LOM	General	Life Cycle	Meta-data	Technical	Educational	Right	Comment	Relation	Classification
Instance	*	*		*	*	*			

D. *The archaeological field instance*

The third field is archaeology. At this level, the standard used for comparison is the CIDOC Conceptual Reference Model (CIDOC CRM) standard. The CIDOC CRM is intended to promote a shared understanding of the cultural heritage information by providing a common and an extensible semantic framework in which any cultural heritage information can be mapped [12]. It is intended to be a common language for the domain experts and implementers to formulate the requirements of the information systems and to serve as a guide for the good practice of the conceptual modeling (see Figure. 10).

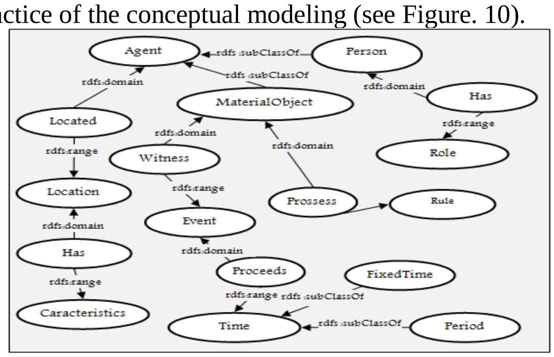


Figure 11. The instance of the archaeological field.

The comparison between the two class diagrams: CIDOC CRM and the proposed model instance lead to the result represented in the following table (see Table V).

TABLE V. THE ARCHAEOLOGICAL FIELD COMPARISON.

CIDOC CRM	Chronological period	Coordinats individual	Appellation	Temporal entity	Material entity	Conceptual	Agent	Dimension	Location	Event	Instance
Instance	*	*	*	*	*	*	*	*	*	*	*

These instances present a large part of each presented standard. The validation of the proposed model requires the validation of these instances. First, in the medical field we have discussed six concepts of DICOM standard, i.e., 55.5% of all concepts. Second, in the educational domain we have used five out of nine concepts in the LOM standard, a percentage amounting to 55.55%.Third, in the museum field, the created instance uses all the concepts of DICOM but not all the sub-concepts (classes) that are almost 100% of concepts. Finally, we have discussed the medical field and we noticed that this instance uses ten classes from the Open EDI standard. The created instances concern different fields. Thus, we mentioned a set of classes from the proposed meta-classes except the classes representing mobility.

V. CONCLUSION

In this work, we presented the various phases of the generic model realization for the pervasive environment. This generic model is presented in a class diagram and it is used in the adaptation phase. Besides, the proposed model is instanced for use in e-learning and NewsML applications through the descriptive languages as RDF, RDFS and OWL. After validating the proposed class diagram model, the instances of this proposed generic model used in the second phase of our proposition which is the adaptation phase. In this phase, we are going to precede three types of adaptation: context adaptation, content adaptation and network adaptation.

VI. REFERENCES

[1] C.Alexander, S. Ishikawa and M. Silverstein, “A pattern language: towns, buildings, construction”, Oxford University Press, ISBN 0195019199, New York, 1977.

[2] D. Salber, A. Dey and G. Abowd, “The Context Toolkit: Aiding the Development of Context-Enabled Applications”, Proceedings of HI’99, Pittsburgh, PA, ACM Press, pp. 434-441, May. 1999.

[3] S. Buchholz, T. Hamann and G. Hbsch, “Comprehensive Structured Context Profiles (CSCP): Design and Experiences”, 2nd IEEE Conference on Pervasive Computing and Communications Workshops PerCom 2004 Workshops 14-17, Orlando, FL, USA. pp. 43-47, March 2004.

[4] H. Chen, F. Perich, T. Finin and A. Joshi, “SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications”, International Conference on Mobile and Ubiquitous Systems: Networking and Services, Key: citeulike:1233279, pp. 258-267, August 2004.

[5] H. Chen, T. Finin and A. Joshi, “The SOUPA Ontology for Pervasive Computing”, Ontologies for Agents: Theory and Experiences, pp. 233-258, 2005.

[6] I. Roussaki1, M. Strimpakou, C. Pils, N. Kalatzis, M. Anagnostou, “Hybrid context modeling: A location-based scheme using ontologies”, Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW’06), pp.6-12, 2006.

[7] M. Kaenampornpan, E. O’Neill and G. Abowd, “Modelling context: an activity theory approach”, EUSAI, Vol. 3295, pp. 367-374, 2004.

[8] Wei Xu, Arta Dilo, Sisi Zlatanova and Peter van Oosterom, "Modelling emergency response processes: Comparative study on OWL and UML ", Joint ISCRAM-CHINA, Gi4DM Conference, Harbin, China, pp. 493-504, August, 2008.

[9] DICOM Standards Committee, Working Group 6, “Digital Imaging and Communications in Medicine (DICOM), DICOM Encapsulation of PDF Documents”, March, 2005.

[10] O. Troulet-Lambert, “UML et XML pour le commerce électronique Business to Business Business to Administration”, HERMES Science Europe, Paris, 2000.

[11] P. Barker, What are IEEE Learning Object Metadata / IMS Learning Resource Metadata?, cetis standards briefings series, 2005.

[12] The official web site: http://www.cidoc-crm.org/, consulted April 2013.

Ontology for Quality Specification in Requirements Engineering

Farideh Heidari, Frances Brazier
System Engineering Section, TBM faculty
Delft University of Technology
Delft, The Netherlands
{f.heidari, f.m.brazier}@tudelft.nl

Pericles Loucopoulos
Department of Informatics and Telematics
Harokopio University of Athens
Athens, Greece
p.loucopoulos@hua.gr

***Abstract*—The field of Requirements Engineering (RE) is arguably one of the most crucial areas in the development of systems in support of organisational structures and processes. Eliciting, negotiating, analysing and validating are RE processes that rely on appropriate abstraction mechanisms. This paper focuses on a specific modelling approach, that of Business Process Modelling (BPM), and the use of a specific ontology for modelling and evaluating quality aspects of business processes. This business process ontology provides an explicit specification of the shared conceptualization and understanding of enterprises between IT and none-IT experts. Specification and measurement of requirements based on an ontology fosters communication between experts. This paper proposes an approach that drives specification and measurement of quality requirements. Application of the proposed approach is illustrated for a simplified version of a business process.**

Keywords- Ontology; Quality requirements; Quality sepcification; Quality measurement; Business process; Business process modelling.

I. INTRODUCTION

Arguably the most significant issue in systems development is getting requirements right and transforming them without information loss into a semantically rich specification, from which various types of software artefacts can be derived [1]. An important challenge for requirements engineering (RE) is to facilitate communication between non-IT and IT specialists [2]. An ontology designed for this purpose with well-defined semantics at the appropriate level of abstraction is the challenge this paper addresses.

Currently, enterprises are described in terms of business processes models. Linking business process quality requirements with business process concepts enables IT and business experts to define their requirements collaboratively at a common abstract level during the earliest stage of design and development of information systems.

The motivation for the work this paper presents is to use ontological constructs to facilitate specification of quality requirements based on business process concepts; in other words, *ontology driven quality requirement specification*. In addition to quality requirements, annotation of business process models with related information artefacts using domains' vocabulary leverages different concepts (goals, rules, patterns, motivation, etc.) into the scope of business process ontology [3, 4]. Each business process modelling (BPM) language provides an ontology consisting of a set of concepts. A systematic realization and representation of concepts and relationships between the concepts of different BPM languages in a business process ontology [5], is essential. The formalism of a generic purpose modelling language (GPML) e.g. UML class diagram provides the ontology description. The approach is exemplified using an example from a real-life business process in an industrial case.

The paper is organized as follows: Section 2 presents a brief summary of related works. Section 3 elaborates on ontologies, in particular business process ontologies and their application and introduces an ontology driven approach to specification and evaluation of quality of business processes. Section 4 illustrates the proposed approach for a business process. The paper concludes in Section 5 with a number of observations, reflections and suggestions for future work.

II. RELATED WORKS

Quality has been the topic of research in several closely related disciplines as requirement engineering, software engineering, workflow analysis, industrial engineering, system dynamics and discrete event simulation. Different levels of granularity can be considered for realizing and measuring quality in an enterprise involving many organizational layers from the very general i.e. organization-wide quality to concepts of business processes. Synoptically, investigation of the most relevant approaches in following aspects will be considered in this section: (A) the way they are being practiced (methodology e.g. systematic or ad hoc), (B) representation of business process and quality requirement (modelling and language dependency), and (C) generalizability of the approach (application) is conducted in this section.

A. Methodlogy

While "focus of work", "required inputs", "expected outputs" and a "set of phases" are prescribed with details in an approach, the approach is considered to be systematic in terms of methodology (e.g. [14], [6], [7], [8], [9]); otherwise

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-266-0

the approach is considered to be ad-hoc in terms of methodology (e.g. [10], [7], [11])

Wolter et al. [12] deploy a method to assign elements of their security model to a process model. Capturing quality dimensions of a business process in form of a framework are considered by Heravizadeh et al. [13]. A framework for evaluation of business process quality is introduced by Kedad et al. [14]. A requirements engineering framework with the aim of allowing active stakeholder participation is introduced by Donzelli et al. [8]. Pourshahid et al. [15] introduce a framework to measure and align processes and goals subjectively. In their work, key performance indicators (KPI) are added to user requirement notation (URN) together with explicit goals for each business process. A scenario-based methodology and a toolset for BPM and analysis is introduced by Glykas[9]. The approach defines and measures KPIs in qualitative as well as quantitative manner.

The approach by Firesmith [7] proposes a checklist of questions over which defects in software-intensive system architectures would be realized. Measurement is included in the structure although the process toward the measurement is not discussed. Lohrmann et al[11] provide a definition for business process quality and introduce business process quality model. There are no details provided on how measurement should be conducted

B. *Modelling and language dependency*

Modelling is concerned with the way an approach represents a business process. The consideration here is the use of formal or semi formal languages in the representation (e.g. [8],[13],[14], [6],[16].) Language dependency examines this fact if an approach's focus is on a specific language. Works by Kedad et al [14] and Said-Cherfi et al. [17] are not tied to a specific modelling language.

Heinrich et al [18] use the quality characteristics and attributes of processes. They distinguish on the basis of the ISO/IEC standard for software quality [10] to enhance BPMN. Saeedi et al [14] propose a set of quality requirement factors for BPMN concepts. Role Activity Diagram notation is considered for representation of business processes by Aburub et al. [16]. The strategic rationale for the choice of business processes to be specified in BPMN models and described in terminology familiar to business people are considered by Decreus et al. [6].

C. *Application*

The application aspect is concerned with the target of the approach. Generic approaches can be applied to all or most situations (e.g. [19], [14]). Specific approaches (e.g. [12], [16]) are dedicated to a particular class or application or business sector.

Wolter et al. [12] focused on security requirements. Aburub et al. [16] introduced an approach in remodelling business processes for identification and inclusion of Non-Functional Requirements (NFRs) for a specific case. With the focus on quality of business process model the approach by Said-Cherfi et al. [17]considers ontologies in a number of specific domains.

There are variations in methodology, in the specification approaches used and the target application of these quality approaches. The desire is to provide an approach, which is systematic and well-structured, generic enough and not tied to a specific domain or situation, and while considering formal expression of business processes is not tied to a particular BPM language. The majority of approaches discussed above are based on the assumption that a formal language (e.g. BPMN) is used to describe business processes, the majority of which use one representation scheme. A few are language independent. Some provide systematic way of working and some are generic enough to be applied in generic situations. The approach introduced in this paper in some ways complements and in others extends existing approaches by emphasizing a well-structured way for specification and objectively measurement of business processes, which is generic in application and language independent.

III. The Ontology Driven Approach

During requirement analysis, an important consideration is to understand current business processes. In this effort, business process modelling plays a key role. Therefore,

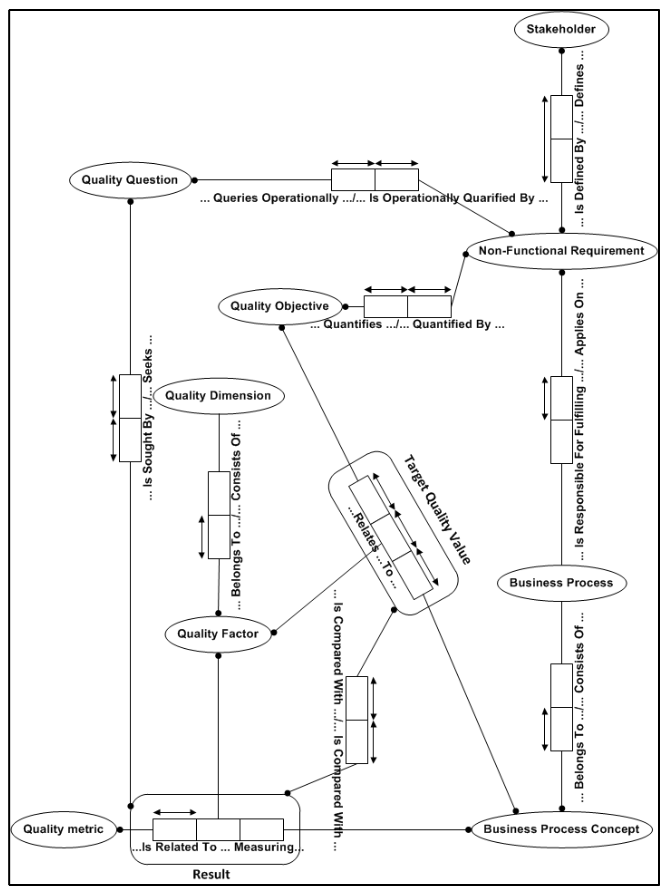


Fig 1. Conceptual Framework for Business Process Concept Quality Specification and Measurement (in ORM)

having an ontology for quality aspects of any business processes, using a corresponding model, would go a long way to unifying the field and to facilitating a more systematic way of treating quality.

Synchronization between requirements and business process models requires a common basis. This common basis can be presented in an ontology confined to requirement aspects. The ontology construction is to provide an explicit representation of knowledge, that can be understood by both computers and people [20].

A business process ontology represents an abstraction of business process concepts, that is universal and not dedicated to one single BPM language. The business process ontology can be designed for business analysts to describe functional as well as non-functional requirements in a single place [21].

Another applications of a business process ontology *confined to quality aspects* is that stakeholders to define their desired requirements in a higher level (meta-model) rather than in specific business process specifications. A business process ontology, *enriched with the desired requirements*, can act as a reference model for future enriched business processes generations.

This ontology can also act as a repository. This repository can have several applications: (a) to represent models created via deploying any of the constructing modelling languages as its instantiations, (b) to be a reference between multiple business process modelling approaches of the same project, (c) to provide the basis for a repository of emerging business process models irrespective of the language used, (d) to be extended to a knowledge base, (e) to facilitate direct implementation, and (f) to be a reference model fostering incorporation of stakeholders' requirements.

This paper proposes an ontological approach to specification and measurement of quality requirement for business process concepts. This approach includes quantitative metrics for business processes in its specifications. The conceptual framework of the approach is shown in Fig. 1. The framework is presented in Object Role Modelling (ORM) notation [22].

The "conceptual framework" encompasses a set of concepts that link requirements to specific business process concepts, their factors and corresponding metrics. Fig.1 depicts that "Requirements", are associated with "Business Processes Concept". A "Requirement" is expressed by a "Stakeholder" and is operationally queried by a set of "Questions". Operationally querying "Requirements" are linked to components that participate in measuring achievement of the "Requirement". "Question" essentially is a query on the 3-nary relationship of "Business Process Concept", "Factor" and "Metric" as depicted in the objectified relationship "Result" related to "Question".

A "Business Process" consists of a set of "Business Process Concepts". A business process "Factor" is an inherent property of a "Business Process" or a "Business Process Concept" that can be measured quantitatively by "Metrics".

The gap between "Objective" and the observed current performance through "Question" is shown in the relationship of "Target" and "Result". Several "Metrics" can be associated to a single "Factor" as there might be several ways for evaluating it. Different stakeholders can indicate different metrics based on their needs [14].

The contribution of this framework is in the establishment of a set of conceptual structures that are independent of descriptive languages, or applications. Applicability of the framework is illustrated for an example of business process in the next section.

In addition to specification and measurement of quality requirements for individual business process concepts, there is a need for measuring requirement fulfilment by a business process as a whole or a part of business process. A business process ontology can foster objective evaluation of the degree to which a quality requirement for a business process is achieved based on achievements of its individual concepts.

To evaluate business processes on the basis of the result of its individual constructs, analysts need to break down process models into more manageable and easily measurable parts. Realization of reusable patterns makes this task more straightforward [23]. A business process ontology defines business process concepts and the relationships among them. In this paper, the five generic patterns of " sequential", " exclusive", "parallel", "loop" and "inclusive" are implemented [5]. Higher degree business process are described as a combination of these patterns [23]. Semantics of individual patterns determine the computation rules and formulas for evaluation of the pattern based on its constructing concepts.

Due to space limitation, discussion on the method, rules and mathematical formulas for measurement (computation) and estimation (prediction) of quality of business processes based on its constructing concepts are not included in this paper. This application of the ontology driven approach is illustrated via an example in the next section.

IV. DEMONSTRATION OF APPLICABABILITY

The applicability of the approach this paper proposes is demonstrated for a simplified version of business process, namely "Accepting clients" from an anonymous enterprise. The business process is known to this enterprise.

A. Evaluation of quality of individual concepts

First, in a more visual way, the instantiation of the framework is provided (Fig. 2). Fig. 2 illustrates not only the business process in terms of a model but also provides examples of the related elements for quality specification and measurement considering the business process concepts. Later, the conceptual framework is instantiated (Fig. 3) to demonstrate its application relates to the example in form of an ORM model.

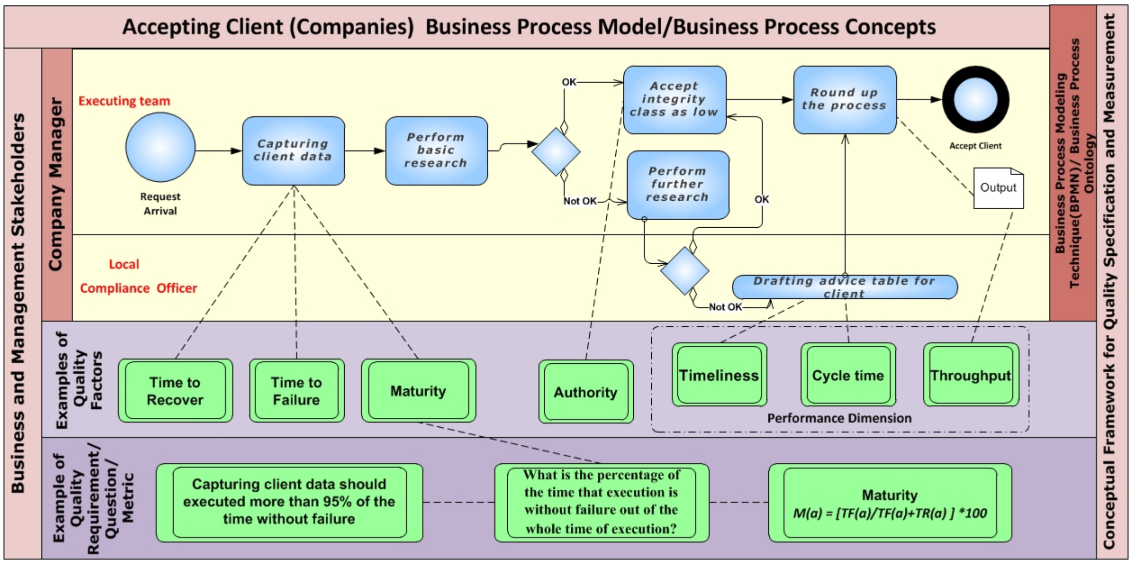


Fig 2. Accepting Client Business Process and Examples of Quality Factor, Requirement, Question and Metric

As can be observed from Fig. 2, there are different departments/roles involved in the process. The process trigger is arrival of a request to accept a client. To accept the client, a set of activities is performed in a predefined order. Some related quality factors are shown in Fig. 2 namely,

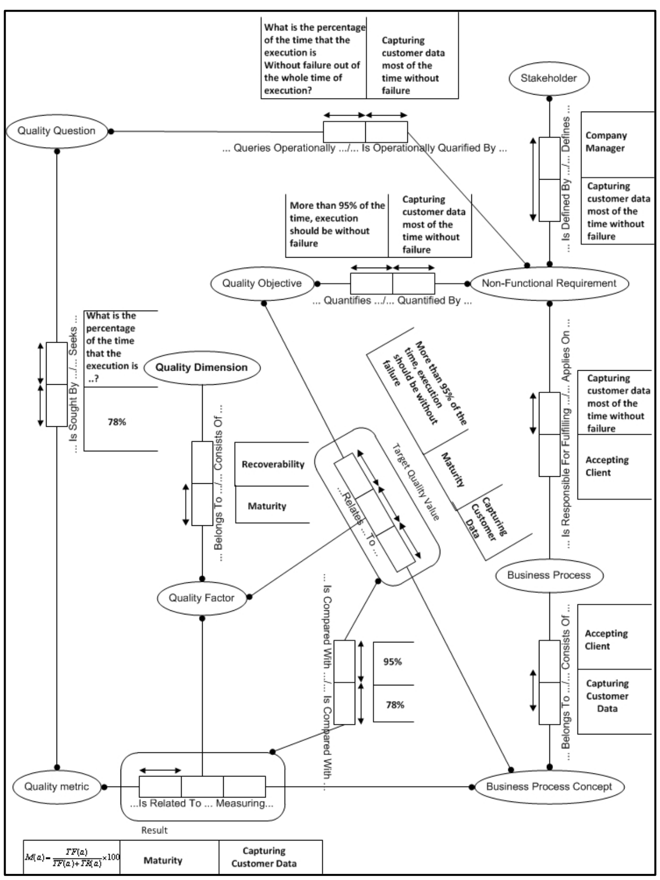


Fig 3. Conceptual Framework Instantiation for Accepting Client Process (Capturing Client Data Should Be Executed More than 95% of the Time Without Failure

time to recover, time to failure, maturity, authority, timeliness, cycle time, and through put. Quality factors are assigned to the business process concepts via dashed lines as shown. For the mater of distinction, quality factors are shown in a separate box below the example. The "business process" is presented via applying BPMN as a "Business process modelling language" supported by a business process ontology e.g. [5].

Fig. 2 shows that the quality requirement of "Capturing client data should be executed more than 95% of the time without failure", is associated with the business processes concept of "Capturing client data"; this concept belongs to business process of "Accepting client". The Requirement is expressed by a the "Company manager" as the stakeholder and is operationally queried by questions of "What is the percentage of the time that execution is without failure out of the whole time of execution?" The quality factor "maturity" can be measured by a quality metric expressed as following:

$$M(a)=[TF(a)/TF(a)+TR(a)]*100 \quad (1)$$

Where "a" demotes the "Activity", TF(a) is the "Time to Failure" and TR(a) is the "Time to Recover".

Applicability of the proposed approach is also demonstrated via instantiation of the conceptual framework (Fig. 3) with regards to the example. The instantiation is focused on quality requirement of " capturing client data should be executed more than 95% of the time without failure". Instances are introduced as "roles" in "fact tables". Information in the fact tables is inline with the example described earlier and provided in Fig 2.

B. *Evaluation of quality of a part of busienss process*

Formula (1) is for calculation of the maturity of a single activity as an individual concept of a business process. Besides single concepts, there is a need for computational formulas enabling one to evaluate quality of a business Measurement of maturity of a part of business process (Fig.

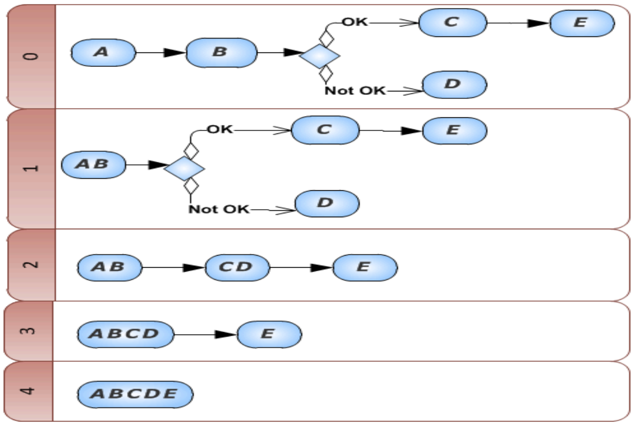


Fig 4. Business Process Reduction Stage

2) executed by "executing team" is considered in this example. The lane under investigation encompasses a flow of activities linked together either with sequential or exclusive relationship; in sequential and exclusive patterns composed of smaller parts.

Inspired by an algorithm offered for workflow reduction in [24], as can be observed in Table 1, the recognized patterns can be reduced to more manageable and easily measurable concepts. Different formulae are introduced for the different patterns. Reduction is performed in different stages. The end result is one single activity (Fig.4). Either estimation or measurement of the activity flow is possible. Based on the reduction patterns and the formulae introduced in Table 1, four steps toward complete reduction are involved in the calculation of maturity of the flow of activity (Fig. 4). For the ease of communication, the name of each activity is replaced simply with just a capital letter. Each phase involves one reduction. The result of reduction of each set of two activities is presented as another activity. The resulted activity name is indicated as the combination of the labels of the constructing activities; for example in stage 1, reduction of sequential patter of "A→B" is a single activity namely "AB" and so on.

For the purpose of simplification, the same value is specified for the maturity measurement and estimation of the value of an activity before reduction. In the examples provided, measurements, assume the client is to be accepted. Probability of activity "C" is 75% and consequently probability of activity "D" is 25%.

TABLE I. REDUCTION FORMULAE

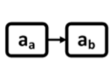
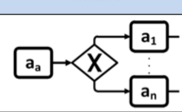
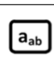
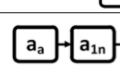
		Pattern		
		Sequential	*Exclusive*	
				Actual
Factor	Calculation			Reduced
Maturity	*Measurement*	$M(ab)=M(a_a)\times M(a_b)/100$	$M(a_{1n})= M(a_i)$	
	Estimation	$M(ab)=M(a_a)\times M(a_b)/100$	$M(a_{1n})=[\sum_1^n p(a_i)\times M(a_i)]/100$	

TABLE II. CALCULATION REULSTS

Activity	Reduction Formula		Result	
	Measurement	*Estimation*	*Measurement*	*Estimation*
A	-----	-----	100	100
B	-----	-----	95	95
C	-----	-----	80	80
D	-----	-----	100	100
E	-----	-----	75	75
AB	M(A) ×M(B)/100		95	95
CD	M(C)	$[M(C)\times 75 + M(D)\times 25]/100$	80	85
ABCD	M(AB) ×M(CD)/100		76	80.75
ABCDE	M(ABCD) ×M(E)/100		57	60.56

Table 2 provides the formulae for calculation of the activities resulting from each of the stages of reduction for two cases of measurement and estimation. The calculation is based on the formula introduced in Table 1.

As demonstrated, calculation of requirement fulfilment of a business process can be based on results of constructing concepts. This provides an objective and valid result for the business process.

V. CONCLUSION AND FUTURE WORK

This paper assumes that the quality of a business process can be defined by the degree to which pre-defined properties of pre-defined concepts identified within a business process are linked to stakeholder requirements. The methodological stance of the approach proposed in this paper is *systematic*. Stakeholder goals are transformed to objectified components, a quality objective and a quality question, that are directly linked to quality factors for a pre-defined business process concept. The approach relies on f*ormal* expressions of business processes in business process models. At the same time, it is *independent* of any language. The utility of approach is *generic*, i.e. applicable to any application and within any domain.

The outcomes of this research are beneficial to the areas of business and management, requirement engineering, software engineering, business process modelling and service-oriented architectures. In the areas of requirement engineering and software engineering, these results make it possible for practitioners to consider quality requirements at the earliest stage.

This paper establishes a strong framework upon which different methodological and technological developments may emerge such as an enhancement of existing business process modelling tools with a simulation component, the development of a workbench for analysing measured qualities and the development of further cases on an industrial basis [25]. Future research will focus on extensions and developments both in theoretical and practical perspectives. Exploring possibilities to enhance existing industrial business process-modelling tools with quality evaluation extensions is currently subject of research. Also, strategic modelling approaches such as system dynamics are to be coupled to business process modelling using parametric definitions according to quality criteria and experimenting with 'what-if scenarios' thus giving stakeholders an early view of the impact of their choices, on the behaviour of a business process [26].

REFERENCES

[1] P. Loucopoulos and V. Karakostas, *System Requirements Engineering*, 1st ed. London: McGraw Hill, 1995.

[2] P. Loucopoulos, "The F3 (From Fuzzy to Formal) View on Requirements Engineering," *Ingénierie des Systèmes d'Information,* vol. 2, pp. 639-655, 1995.

[3] P. Loucopoulos and E. Katsouli, "Modelling Business Rules in an Office Environment," *SIGOIS Bulletin,* vol. 13, pp. 28-37, 1992.

[4] E. Kavakli and P. Loucopoulos, "Experiences with Goal-Oriented Modelling of Organisational Change," *IEEE Transactions on Systems, Man and Cybernetics - Part C,* vol. 36, pp. 221-235, 2006.

[5] F. Heidari, P. Loucopoulos, and Z. Kedad, "A quality-oriented business process meta-model," in *International Workshop on Enterprise & Organizational Modelling and Simulation,CAiSE/EOMAS 2011, Lecture Notes in Business Information Processing*, 2011, pp. 85-99.

[6] K. Decreus and G. Poels, "A Goal-Oriented Requirements Engineering Method for Business Processes," in *CAiSE Forum*, Berlin Heidelberg, 2010, pp. 29-43.

[7] D. Firesmith, "Quality Requirements Checklist," *Journal of Object Technology* vol. 4, pp. 1-8, 2005.

[8] P. Donzelli and P. Bresciani, "Improving Requirements Engineering by Quality Modelling – A Quality-Based Requirements Engineering Framework," *Journal of Research and Practice in Information Technology,* vol. 36, pp. 277-294, 2004.

[9] M. M. Glykas, "Effort Based Performance Measurement in Business Process Management," *Knowledge and Process Management,* vol. 18, pp. 10-33, 2011.

[10] ISO/IEC, "Software engineering - Product quality - Part 4: Quality in use metrics," ISO, Switzerland2004.

[11] M. Lohrmann and M. Reichert, " Understanding Business Process Quality," in *Business Process Management ,Theory and Applications. Studies in Computational Intelligence (444), 41-73*, ed Berlin Heidelberg: Springer 2013, pp. 41-73.

[12] C. Wolter, M. Menzel, A. Schaad, P. Miseldine, and C. Meinel, "Model-driven business process security requirement specification," *Journal of Systems Architecture,* vol. 55, pp. 211-223, 2009.

[13] M. Heravizadeh, J. Mendling, and M. Rosemann, "Dimensions of Business Processes Quality (QoBP)," in *Proceedings of the 6th International Conference on Business Process Management Workshops (BPM Workshops)*, Milan, 2008, pp. 80-91.

[14] Z. Kedad and P. Loucopoulos, "Considering quality factors for business processes during requirement engineering," in *Fifth International Conference on Research Challenges in Information Science (RCIS)*, 2011, pp. 1-9.

[15] A. Pourshahid, D. Amyot, L. Peyton, S. Ghanavati, P. Chen, M. Weiss*, et al.*, "Towared an integrated user requirement notation framework and tool for business process management," in *3rd Int. MCeTech Conf. on eTechnologies*, Montréal, Canada, 2008, pp. 3-15.

[16] F. Aburub, M. Odeh, and I. Beeson, "Modelling non-functional requirements of business processes," *Information and Software Technology,* vol. 49, pp. 1162-1171, 2007.

[17] S. Said-Cherfi, S. Ayad, and I. Comyn-Wattiau, "Aligning Business Process Models and Domain Knowledge: A Meta-modeling Approach " in *Advances in Databases and Information Systems, AISC 186*, T. Morzy, T. Hӓrder, and R. Wrembel, Eds., ed Berlin, Heidelberg: Springer, 2013, pp. 45-56.

[18] R. Heinrich, A. Kappe, and B. Paech, "Modeling Quality Information within Business Process Models," *Proceedings of the SQMB'11 Workshop, TUM-I1104, pp. 4-13, Karlsruhe (Germany), February,* pp.4-13, 2011.

[19] P. Loucopoulos and F. Heidari, "Evaluating Quality of Business Processes," in *Modelling and Quality in Requirements Engineering, Essays Dedicated to Martin Glinz on the Occasion of His 60th Birthday*, N. Seyff and A. Koziolek, Eds., ed Munster: MV-Wissenschaft, 2012, pp. 61-73.

[20] S. Z. Fan, L. ; Sun, Z. , "An ontology based method for business process integration " in *International Conference on Interoperability for Enterprise Software and Applications*, China, 2009, pp. 135-139.

[21] D. E. Jenz, "Business processs ontologies: speeding up business process implementation," Jenz & Partner GmbH2003.

[22] T. A. Halpin, "Object-Role Modeling: an overview. Online at http://www.orm.net," 1997.

[23] T. Gschwind, J. Koehler, and J. Wong, "Applying Patterns during Business Process Modeling," in *BPM 2008. LNCS*. vol. 5240, M. Dumas, M. Reichert, and M. C. Shan, Eds., ed Heidelberg: Springer,, 2008, pp. 4–19.

[24] J. Cardoso, J. Miller, and J. Arnold, "Modeling quality of service for workflows and web service processes," vol. #02-200, ed: LSDIS Lab, Computer Science, University of Georgia, 2002.

[25] D. Karagiannis, "A Business process Based Modelling Extension for Regulatory Compliance," presented at the Multikonferenz Wirtschaftsinformatik 2008, 2008.

[26] P. Loucopoulos, K. Zografos , and N. Prekas, "Requirements Elicitation for the Design of Venue Operations for the Athens2004 Olympic Games," in *Proceedings of 11th IEEE International Requirements Engineering Conference*, Monterey Bay, California, U.S.A, 2003, pp. 223-232.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-266-0

Ontological Representation of Public Web Services

Maricela Bravo
Systems Department
UAM-Azcapotzalco
DF, Mexico
mcbc@correo.azc.uam.mx

Mónica Silva-López
Systems Department
UAM-Azcapotzalco
DF, Mexico
misl@correo.azc.uam.mx

Blanca Silva-López
Systems Department
UAM-Azcapotzalco
DF, Mexico
rbsl@correo.azc.uam.mx

***Abstract*—Among the main benefits of service-oriented architectures is the reutilization of software components that may solve specific tasks for complex problems, requiring the composition of multiple Web services. Currently Internet is largely populated with Web services offered by different providers and published in various Web repositories. However, public available Web services still suffer from problems that have been widely discussed, such as the lack of functional semantics. This lack of semantics makes very difficult the automatic discovery and invocation of public Web services, even when the system integrator can obtain a copy of the WSDL file. This paper describes an ontological approach for discovering similarity relations between public Web services. The objective of this work is to extract relevant data that is coded into service operations descriptions, calculate similarity measures between them, represent the discovered similarities in an ontological form, and execute inference. Experimental results show that the overall process towards the automation of public Web services discovery based on ontology population and structural similarity measures is feasible and can be completely automated.**

Keywords-Web services; Structural Similarity Measures; Similarity Relations Discovery; Automated Ontology Population; and Inference.

I. INTRODUCTION

In the last decade, many software vendors have developed, deployed and offered software as services using interface description languages, such as the Web Service Description Language (WSDL) [1]. In order to make their services available online, providers publish their service descriptions in public Web service repositories, which may or may not be conformant to a specific standard such as UDDI [2] or ebXML [3]. When software integrators search for Web services that meet certain criteria in public repositories, and try to select and invoke existing Web services, they may face some of the following problems:

Lack of well-documented Web service descriptions. This is a common problem that many public Web service clients or requestors face. A study and report of this problem was presented by Rodríguez et al. [4]. In this work, authors identify common mistakes in WSDL documents: inappropriate or lacking comments, use of ambiguous names for the main elements, redundant port-types, low cohesive operations in the same port-type, enclosed data model, redundant data models, etc. According to [4], less than 50% of the studied WSDL files have some documentation. Additionally, the naming of services, operations, messages and parameters does not follow any convention, and there is no obligation to provide additional semantic information. These reasons cause enormous difficulties during search, selection and invocation of services.

Lack of semantically enhanced Web services repositories. There are many public Web service repositories, but they do not offer sufficient semantic information about the service functionality, making very difficult the automated exploitation of deployed Web services. Majority of public repositories offer key-based search mechanisms, and some sort of classifications, but none of them offer correlations discovery between existing services considering provided interface (template) information.

Different solutions have been proposed to solve these limitations. The semantic Web has influenced many works by providing logic-based mechanisms to describe, annotate and discover Web services. Within this context, McIlarith et al. [5] proposed one of the first initiatives to markup Web services based on DAML (ontology language), which started the important research area of "Semantic Web Services". The term Semantic Web Services is related to the set of technologies and models based on the implementation and exploitation of ontologies as a mechanism to semantically enhance service descriptions, for instance: OWL-S [6], WSMO [7], and SAWSDL [8]. However, these methods require human experts intervention to construct ontologies and annotate Web service descriptions before their deployment.

From the perspective of Web services providers, if they want to take advantage of these semantic-based technologies, they will have to re-design their solutions with the following considerations: in case of annotating semantically their Web services using SAWSDL, they need to construct or select an ontological representation relative to the domain of the services offered; in the case of using OWL-S, service providers need to learn this model and use the tools available to create the corresponding ontological descriptions of their services; and in case of using WSMO, the learning curve is steep because it requires more effort to understand and use the complex framework of WSMO with a new ontology language. Considering that service providers are familiar with software development, but not necessarily with ontologies or semantic Web technologies.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-266-0

From the point of view of a service requestor, the first problem that he will face is to find Web services repositories containing semantic Web service descriptions using these technologies. Furthermore, he will not find a universal repository containing all types of semantic service descriptions. Until now there is no reported approach or tool that automatically translates or connects (without human intervention) any pre-existing Web service description to any of the aforementioned solutions.

Despite the increasing popularity of semantic-based technologies, the numerous researchers devoted to them, the great advances and achievements; there is still an important gap between these semantic-based Web service technologies and the pre-existing Web services, which were deployed using only WSDL (including the common mistakes pointed by Crasso). There is no doubt that the semantic Web trend will continue and will consolidate in the following years, and if service providers want to stay competitive, they need to adapt and re-deploy their services using these technologies. Meanwhile, from the perspective of computer science, many solutions can be designed and constructed to overcome these difficulties.

The solution reported in this paper relies on the following basis, considering that Web services can be described using different languages, there are essential common elements that all Web services description languages must provide: a general communication interface that the client uses to create a proxy object to invoke the service remotely. This communication interface must describe information about the functions that the service offers (operation in WSDL, profile in OWL-S or capability in WSMO) as well as the correct description of input and output parameters. Taking into account these common elements between services described in any of these languages and that ontologies represent the cutting edge technological movement, this article describes an ontological representation for public Web services. With this motivation in mind, this paper describes two contributions:

A service mining process which extracts relevant service elements coded into service descriptions, calculates similarity measures, and discovers semantic relationships between them. In this work, the discovery of similarity relations is based on a set of structural and syntactical similarity measures.

An ontology-based representation of public Web services, which serves as a service repository which allows the dynamic acquisition of more service instances and discovery of similarities among them. This service ontology allows the definition of query rules to support complex service tasks such as search, discovery, selection, substitution and composition. An additional benefit of using an ontology-based representation of discovered similarities between Web services is the possibility of inter-connection and inter-operation with existing semantic models.

The rest of the paper is organized as follows: in Section 2, the Web service mining approach is described; in Section 3, experimentation is presented; in Section 4, useful application scenarios are described to show the applicability of this work; finally, in Section 5, conclusions are presented.

II. RELATED WORK

Web service automatic mining is the task of searching (by means of crawlers), retrieving and parsing public Web services. Research topics related with Web service mining are data mining and knowledge discovery. Hamel et al. [9] describe Web service mining as the process of applying data mining techniques on Web service logs, with the objective of discovering actionable Web service intelligence. In particular authors analyze the requirements to deal with four mining levels: the interface level, the abstract process level, the choreography level and the orchestration model of a composite Web service level. However, this work reports this analysis result with no experimentation or real implementations of the four mining levels. Among the main difficulties that they faced is “the lack of existing public Web service execution logs” to work with.

The work reported in Chen et al. [10] is closely related with this one, they use a bottom up discovery approach for mining Web services, use an ontology to represent discovered relations and a set of rules to obtain more definitions between Web services. However, they do not provide any experimental evidence of "real world" Web services.

A service mining framework is reported by Zheng and Bougettaya in [11]. In this work, authors describe a bottom up approach framework for mining Web services. Their main focus is on discovering any interesting and useful service composition that may came up during the mining process, with no goal containing specific search criteria. In contrast, the work reported in current paper has a similar intention mining process, in the sense that no objective criteria is provided, but the semantic relations discovered are not tailored only for composition purposes.

Zhang et al. [12] report a composite Web services discovery technique based on community mining. In this work, authors address the problem of finding services that are more suitable for a common goal to complete a task. In particular, they propose a method of mining the service community by exploiting service execution logs. The main difference between Zhang work and the approach described in current paper is the information source of the mining process.

Young-Ju et al. [13] argue that there exist a vast number of public available services that cannot be utilized by tools that enable service users to create mashups without programming knowledge. They propose a solution based on the combination of existing description languages and learning ontology mechanisms in order to enable the development of semantic web services compliant with architectural style of RESTful web services. Their ontology learning mechanism is used to extract and cluster service parameters, producing a parameter-based domain ontology which is evaluated against a traditional keyword-based service search mechanism.

Yousefipour et al. [14] propose an ontology-based framework for the discovery of semantic Web services (SWS) using a QoS approach. Their framework describes an ontology manager component which handles the provider and the requester domain or general ontologies. This component merges these ontologies with general ontologies and creates a new generalized ontology, which is used for ranking the resulting list of SWS. Even though authors address automatic discovery of SWS by means of ontologies, their ontologies are domain-oriented. In contrast, in this paper ontologies are used for modeling service programmatic interfaces aiming at supporting automatic search and discovery of public available Web services.

Yoo Jung et al. [15] address the problem of annotating Web services from the Deep Web. Deep Web refers to Web pages that are not accessible to search engines. In particular, authors consider Web forms interface pages as Deep Web services that reflect the real content types of the Deep Web. Their proposed solution consists of the automatic generation of a domain ontology for semantic annotation of Web services. Such domain ontology is built based on Web page attributes (any items of descriptive information about the site). Their research main goal is improving the automatic search and discovery of public Web services. However, their service description sources are different as they are using Web form interface pages instead of a formal service description language.

Sabou and Pan [16] presented a study of the major problems with Web service repositories (some of them are no longer available, however the result of the study is still relevant). They concluded that Web service repositories use simple techniques of accessing the content of Web services, browsing across services listings relies on few and low quality metadata, and metadata is not fully exploited for presentation. Authors also proposed various semantic-based solutions to enhance semantically service repositories. Retaking the early ideas of these authors, the solution that is reported in this article is to lay the foundations for the automatic construction of public Web services repositories based on ontologies.

Comparison with related work. The idea of retrieving , clustering and mining Web services using a semantic approach is no new. However, none of reported works have fully achieved the level of automation with real existing Web services. Some works do not offer experimentation with real world service implementations [9], [10]. Service mining-related works were not designed with a global vision (considering programmatic interfaces and application domains) to support all service tasks; for instance [11] describes an approach tailored only for composition, [13] presents the construction of ontologies with a specific parameter-domain approach, or application-domain ontologies [14]. The rest of work use different service information sources: service execution logs [12], and Web form interface pages instead of a formal service description language [15]. The approach reported in this paper aims at automating mining real world Web services to support all service tasks based on programmatic interface ontologies and domain ontologies using as a source public Web service descriptions.

III. MINING PUBLIC WEB SERVICES

In this paper, Mining Web Services is defined as the task of unveiling similarities that hold across multiple Web services descriptions; in order to support complex tasks, such as: discovery, selection, matchmaking, substitution and composition of Web services. Data and text mining have the main purpose of discovering patterns from data and produce new information or knowledge. In this context, the objective of mining public Web services descriptions is to find certain patterns or relationships (patterns and relationships are used interchangeable as synonyms) based on a set of similarity measures.

The process of mining Web services is depicted in Figure 1. This process involves the following phases:

Retrieving public Web services. This phase consists of searching and copying service descriptions files from the Web. The objective is to gather information about services available on the Web and maintain a local repository of retrieved public Web services. To achieve this objective, a common strategy is to program several softbots that seek for services on the entire Web; or visit specific service repositories that manage lists of services. One of the most representative public Web service repositories is Seekda [17] search engine, which currently catalogs more than 28,000 service descriptions.

Parsing Web services descriptions. This phase consist of reading description files, identify the relevant elements, retrieve and process them. With this regard, two important requirements have to be addressed: heterogeneity of service description languages (SDL) and selection of the relevant data to be retrieved from service files. The former requirement is derived from the existence of various SDLs: WSDL, OWL-S, WSML, and SAWSDL. Both requirements are addressed by identifying the essential common elements that all SDLs must provide: a communication interface that the client uses to create a proxy object to invoke the service remotely. This communication interface must describe information about the functions that the service offers (operation in WSDL, profile in OWL-S or capability in WSML) as well as the descriptions (name and data types) of input and output parameters.

Similarities discovery, during this phase a set of similarity measures are calculated to enable the generation of new relationships between services and entail new knowledge about these relations. During this phase the mining process can be configured through the combination of various similarity measures in order to find more interesting results depending on the user needs and application objective.

Inference and maintainability. This phase consist of the execution of a set of inference rules which generate and maintain similarity relations into the ontology whenever new service instances are added. This phase also allows the definition of more rules to construct interesting relations based on the basic similarity relations. For instance, if there are two service functions which hold input and output parameter similarities, and also hold a semantic similarity on their function names, then a combined similarity can be defined to establish a structural similarity (covering parameter and function names). Another possibility is to offer mechanisms of dependency checking if a correlated service instance is deleted.

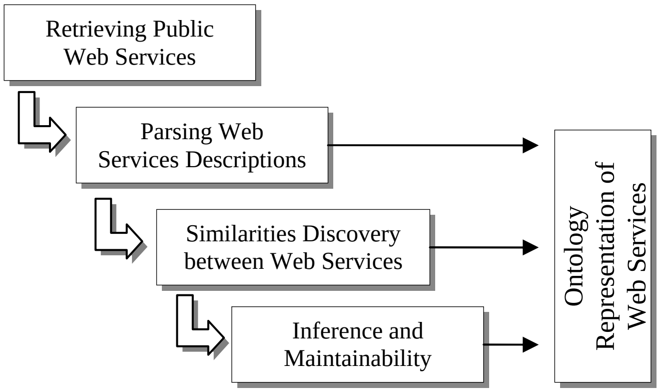


Figure 1. The process of mining public Web services.

An important element of this mining process is the ontological representation, which consists of an ontology management interface and the resulting ontology. The Ontology management is a programming interface through which the mining phases read, write and update concepts and semantic relations into the ontology.

The Web service Ontology is a formal and logical representation of the mined services together with all semantic relationships.

IV. ONTOLOGY TO REPRESENT WEB SERVICES

In 1993, Gruber defined Ontology as "an explicit specification of a conceptualization" [18]. The Web service ontology described in this paper, aims at providing fundamental inter-relations representation of Web service core concepts (functions, input parameters and output parameters). The main entities (classes) are *Service*, *Function* and *Parameter*. Figure 2 shows a general view of this ontology and its interrelationships. The *Parameter* class is sub-classified into *InputParameter* and *OutputParameter* classes.

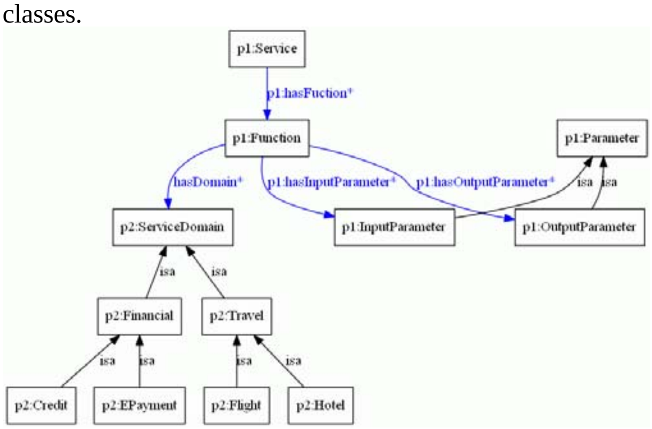


Figure 2. Ontology for the representation of Web services and their related application domains.

Data type properties were defined as follows. For class ***Service*** *hasServiceName* and *hasURL* data type properties were defined to take only *xsd*:*string* data values. For class ***Function***, the *hasFunctionName* data type property was established, allowing only *xsd:string* data values. For class ***Parameter***, the *hasParameterDataType* and *hasParameterName* data type properties were defined to take *xsd:string* data values. Identification of semantic relationships between individuals of different classes in the ontology is implemented as object properties. *Service* class relates with *Function* class through *hasFunction* object property. Restrictions to this property are that a service has at least one function, and can have many functions. *Function* class is related with class *Parameter* through the *hasInputParameter* and *hasOutputParameter* object properties. Object property *hasInputParameter* is restricted to take values only from the class *InputParameters*; likewise, *hasOutputParameters* object property takes values only from the *OutputParameters* class.

V. SIMILARITY MEASURES

Based on the work reported in Bravo and Alvarado [19], in this section various structural and syntactic similarity measures are described.

A. Function name similarity

Let $Oname_1$ and $Oname_2$ be two compound function names from two different Web services. $Oname_1$ consisting of a set of lexical tokens identified by $OnameTokens_1$. $Oname_2$ consisting of a set of lexical tokens identified by $OnameTokens_2$.

The lexical similarity between names is calculated using the Jaccard Similarity Coefficient:

$$\textbf{\textit{FunctionNameSim}}(Oname_1, Oname_2) = |OnameTokens_1 \cap OnameTokens_2| \,/\, |OnameTokens_1 \cup OnameTokens_2| \quad (1)$$

The ***FunctionNameSim*** similarity measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity between both function names, and a returned value of 0 represents a total difference between names.

B. Input parameter similarity

Let $O_1 = (Oname_1, Ip_1)$, $O_2 = (Oname_2, Ip_2)$ be two functions from different Web services, with $Oname_i$ representing the function name and Ip_i the set of n input parameters described as follows:

$Ip_1 = \{ (nameP_1, typeP_1), (nameP_2, typeP_2), \dots, (nameP_n, typeP_n)\}$,

$Ip_2 = \{ (nameP_1, typeP_1), (nameP_2, typeP_2), \dots, (nameP_n, typeP_n)\}$.

Each parameter is defined by a pair of name and data type (*nameP*, *typeP*). The input parameter similarity is calculated as follows:

$$\textbf{\textit{InputParSim}}(O_1, O_2) = |Ip_1 \cap Ip_2| \,/\, |Ip_1 \cup Ip_2| \quad (2)$$

The ***InputParamSim*** measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

C. Output parameter similarity

Similarly to (1), a measure to evaluate the lexical similarity between output parameter **names** is defined. Let $OPname_1$, $OPname_2$, be two output parameter names from different Web service functions, each consisting of a set of lexical tokens identified by $OPnameTokens_1$ and $OPnameTokens_2$, respectively. The output parameter name lexical similarity is calculated by:

$$\boldsymbol{OPnameSim}(OPname_1\,,\,OPname_2) = \left|OPnameTokens_1 \cap OPnameTokens_2\right| \;/\; \left|OPnameTokens_1 \cup OPnameTokens_2\right| \qquad (3)$$

The ***OPnameSim*** measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

Let $OPtype_1$, $OPtype_2$, be two output parameter data **types** from different Web service functions. The output parameter data type similarity between them is calculated as follows:

$$\boldsymbol{OPtypeSim}(OPtype_1\,,\,OPtype_2) = \begin{cases} 1, & \text{if } OPtype_1 = OPtype_2 \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

The ***OPtypeSim*** measure will return a value of 1 if both types are equal, and a value of 0 if they are different.

D. Average output similarity

Let $O_1 = (Oname_1, Op_1)$, and O_2, $= (Oname_2, Op_2)$, be two functions from different Web services, with name $Oname_i$ and the output parameter object Op_i of function i. Each output parameter object Opi consists of a pair of name and data type, $Op_1 = (OPname_1, OPtype_1)$, and $Op_2 = (OPname_2, OPtype_2)$. Particularly, O_1 and O_2 are output equivalent if $((OPname_1 = OPname_2)$ and $(OPtype_1 = OPtype_2))$.

The output parameter similarity is calculated as the average of output parameter name similarity and output parameter data type similarity as follows:

$$\boldsymbol{OutputParSim}(O_1\,,\,O_2) = [\, OPnameSim(OPname_1\,,\,OPname_2) + OPtypeSim(OPtype_1\,,\,OPtype_2)\,] \,/\, 2 \qquad (5)$$

The ***OutputParSim*** measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

E. Structural similarity

Structural similarity represents the average of parameter name similarity (1), input parameter similarity (2) and output parameter similarity (5). Let $O_1 = (Oname_1, Ip_1, Op_1)$, and $O_2 = (Oname_2, Ip_2, Op_2)$, be two Web service functions with their respective sets of input and output parameters; the level of structural similarity between them is calculated as follows:

$$\boldsymbol{StructuralSim}(O_1\,,\,O_2) = [\, \boldsymbol{FunctionNameSim}\,(Oname_1\,,\,Oname_2) + \boldsymbol{InputParSim}(O_1\,,\,O_2) + \boldsymbol{OutputParSim}(O_1\,,\,O_2)\,] \,/\, 3 \qquad (6)$$

The ***StructuralSim*** measure will return a value in the range [0, 1], where a returned value of 1 represents a total similarity, and a value of 0 represents a total difference.

Further similarity measures can be defined and combined to obtain more interesting similarity results between services.

VI. EXPERIMENTATION

For experimentation 37 public Web service descriptions (WSDL) files were retrieved from Seekda [17]. The architecture depicted in Figure 1 was implemented as follows: a Web service data extraction module, which browses any set of public available WSDL files and extracts the service name, the set of function names, the names and data types of input and output parameters; an ontology population module, which registers into the ontology new function instances after data is extracted from WDSL files, and a similarity relations discovery module, which calculates structural similarities between function pairs and registers new semantic relations between compared individuals, if the level of similarity resulted higher than a threshold.

The parser module extracted the function names, input and output parameter names and types from the initial 37 Web service description files. As a result, the ontology was populated with a total of 537 new *Functions*, and 6317 *Parameters*: 3155 *InputParameters* and 3162 *OutputParameters*.

The discovery of structural similarity relations was calculated between all individuals from the *Function* class. Resulting relations were named *isFunctionNameSimilarTo*, *isOutputParamSimilarTo*, *isInputParamSimilarTo*, and *isStructuralSimilarTo*. If the resulting level of similarity is higher than an established threshold, then a similarity relationship is generated between both functions in the ontology. For the set of 537 functions, similarity results are shown in Table 1. Thereafter, the ontology continues growing in similarity relationships as new services are registered. In this case, the resulting ontology is considered dynamic and evolving over time.

TABLE I. NUMBER OF DISCOVERED RELATIONS

Similarity relationship	**Total**
Function name similarity	213
Input parameter similarity	470
Output parameter similarity	1440
Structural similarity	184

The set of structural similar relationships is a combined result of the function name, input parameter and output parameter similarities for each function pair, which is the main reason of the reduced number of relations in comparison with the three previous.

VII. APPLICATION CASES

Searching and discovery of specific service functionalities are among the most important service-related tasks, because it allows software developers and integrators to find specific services functionalities which satisfy their needs. Majority of Web service repositories offer basic search mechanisms, mostly based on key-word and service category matching. The service ontology reported in this paper supports the same search mechanism, but the set of similarity relations discovered and established between services functions; allow seeking and finding more services functions that are structurally related, returning more and significant functions. The following rules are specified to query the ontology and obtain answers about the set of functions being treated. The query rule showed in (7) displays pairs of *Functions* instances for which a relation of *Input Parameter* similarity was discovered and established.

$$Function(?x) \wedge Function(?y) \wedge isInputParamSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (7)$$

Similarly, the query rule shown in (8) displays pairs of *Functions* instances for which a relation of *Output Parameter* similarity was discovered and established.

$$Function(?x) \wedge Function(?y) \wedge isOutputParamSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (8)$$

The query rule showed in (9) displays pairs of Functions individuals for which a relation of Function Name similarity was discovered and established.

$$Function(?x) \wedge Function(?y) \wedge isFunctionNameSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (9)$$

Finally, the query rule presented in (10) is very useful because it allows inferring what functions may be ***substitutable*** each other, provided they meet three conditions: Input Parameter similarity, Output Parameter similarity and Function Name similarity.

$$Function(?x) \wedge Function(?y) \wedge isInputParamSimilarTo(?x, ?y) \wedge isOutputParamSimilarTo(?x, ?y) \wedge isFunctionNameSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (10)$$

Rule (11) searches flight service functions which return flying routes. Results of this query-rule are shown in Table 2.

$$FlightServices(?x) \wedge hasOperation(?x, ?y) \wedge hasOperationName(?y, ?str) \wedge swrlb{:}contains(?str, \text{"}Route\text{"}) \rightarrow sqwrl{:}select(?x, ?y) \quad (11)$$

TABLE II. SERVICES THAT OFFER FLYING ROUTES FUNCTIONS

Service	Function
Volagratis	Volagratis-getRoutes
Arc	Arc-GetRoutes

Using the service ontology it is possible to extend the search of flight functions using the input parameter similarity relation. Rule (12) searches flight service functions which return flying routes and similar functions which hold an *Input Similarity* relationship. Result is shown in Table 3.

$$Function(?x) \wedge hasOperationName(?x, ?\text{str}) \wedge swrlb{:}contains(?\text{str}, \text{"Route"}) \wedge Functions(?y) \wedge isInputParamSimilarTo(?x, ?y) \rightarrow sqwrl{:}select(?x, ?y) \quad (12)$$

TABLE III. SERVICES THAT OFFER FLYING ROUTES FUNCTIONS

Function	Function
Arc-GetRoutes	Arc-GetAvailability

Another example of searching "booking" functions using the function name similarity relation is executed with the query rule (13). Results of this query are shown in Table 4.

$$Function(?\text{x}) \wedge hasOperationName(?\text{x}, ?\text{str}) \wedge swrlb{:}containsIgnoreCase(?\text{str}, \text{"booking"}) \wedge Function(?\text{y}) \wedge isOperationNameSimilarTo(?\text{x}, ?\text{y}) \rightarrow sqwrl{:}select(?\text{x}, ?\text{y}) \quad (13)$$

TABLE IV. SERVICES THAT OFFER FLYING ROUTES FUNCTIONS

Function	Function
Hotelmercado_WS-SetConfirmBooking	Hotelmercado_WS-ConfirmBooking
Hotelmercado_WS-GetBookingInfo	Hotelmercado_WS-GetProcBookingInfo
TourConexWebService-doMainServiceHotelBooking	TourConexWebService-doMainServiceCarBooking
TourConexWebService-doBookingStatistic	TourConexWebService-doBooking
pegas-cancelBooking	MORSWebService-CancelTransferBooking
pegas-cancelBooking	MORSWebService-CancelHotelBooking
TourConexWebService-doCarBooking	TourConexWebService-doMainServiceCarBooking
TourConexWebService-doCarBooking	TourConexWebService-doBooking
pegas-confirmBooking	Hotelmercado_WS-SetConfirmBooking
pegas-confirmBooking	Hotelmercado_WS-ConfirmBooking
TourConexWebService-doBooking	TourConexWebService-doErvBooking
TourConexWebService-doBooking	TourConexWebService-doTicketsafeBooking
TourConexWebService-doBooking	TourConexWebService-doMultiBooking

Substitution is another important task for the Web service community; it allows searching and selecting a similar service function that matches input and output parameters. The query-rule (14) applied to the service ontology helps the developer to search and find "substitutable" services, based on syntactic and structural similarity measures. Results of this query are shown in Table 5. A normal and common service repository does not support such kind of searches. The developer should do so manually, requiring more effort and time.

$$\begin{aligned}&\mathit{Function}(?x)\ \wedge \\ &\quad \mathit{Function}(?y)\ \wedge \\ &\quad \mathit{isInputParamSimilarTo}(?x, ?y)\ \wedge \\ &\quad \mathit{isOutputParamSimilarTo}(?x, ?y)\ \wedge \\ &\quad \mathit{isOperationNameSimilarTo}(?x, ?y)\ \wedge \\ &\quad \mathit{hasOperationName}(?x, ?str1)\ \wedge \\ &\quad \mathit{hasOperationName}(?y, ?str2)\ \wedge \\ &\quad \mathit{swrlb{:}notEqual}(?str1, ?str2) \\ &\quad \rightarrow \mathit{sqwrl{:}select}(?x, ?y)\end{aligned} \tag{14}$$

TABLE V. SUBSTITUTABLE SERVICE FUNCTIONS

Function	Function
BookingLand-CountryProvinceList	BookingLand-CountryProvinceCityList
CHotelsWebService5-startTransaction	CHotelsWebService5-startTransactionMulti
WSNewHotelSrv-GetSimpleAvailability	WSNewHotelSrv-GetSimpleAvailabilityTeste
MORSWebService-Ping	CreditCardServiceV1-ping
WSNewHotelSrv-MakeSimpleReservationTeste	WSNewHotelSrv-MakeSimpleReservation
pegas-getSpecifiedFlightList	pegas-getFlightList
WSNewHotelSrv-DeleteUserHotel	WSNewHotelSrv-DeleteHotel
WSNewHotelSrv-DeleteUserHotel	WSNewHotelSrv-DeleteUser
BookingLand-ProviderAvailabilityEx	BookingLand-ProviderAvailability
BookingLand-ProviderSearchQuick	BookingLand-ProviderSearchQuickEx

VIII. PERFORMANCE ANALYSIS

Performance analysis of service-related tasks is an important issue whenever these tasks are based on ontological representation. In particular, in this paper the following service tasks are of performance concern:

Ontology population. This is the most time consuming task because for each service instance treated requires the execution of two operations: *service parsing* and *service ontology recording*. Which means that for each service, the parser extracts its operation names and respective input and output parameters, and then records all instances into their ontology classes. In particular, for the set of 37 initial *Services* used for experimentation, a total of 537 *Functions*, and 6317 *Parameters* were registered into the ontology file. Therefore, this task required a total of 37 service parsing operations and the sum of 37 + 537 + 6317 = 6891 ontology-write operations. Obviously, the more service instances are treated the more time is needed. However, this particular time-consuming task is not considered as critical, because it is executed only once per service set. Even more, when new service instances are to be recorded into the same ontology, they are first validated for non redundancy, therefore only new different services are allocated. Ontology population is a time-consuming task, but is not a frequent task.

Search, discovery and substitution. In a traditional implementation approach these tasks would require traversing the entire ontology T-Box and A-Box to find particular class instances, relation instances or individuals. However, in this paper the use of a rule language enhanced with querying constructs (SWRL) allows the definition and execution of rule-based search, discovery and substitution. A rule-based querying mechanism offers improved performance, as it filters only the necessary class, relations, axioms and individuals needed for the execution of each rule. For instance, when the inference engine executes the query rule (7) it requires to load a total of 537 *Function* instances and 470 *InputParameterSimilarity* relations, resulting in a space-reduced selection operation.

IX. CONCLUSIONS AND FUTURE WORK

Results show advances on automatic similarity relations discovery from public available Web service descriptions. The automatic population of the ontology with existing WSDL files is a relevant advance towards the automated reutilization and construction of service-based solutions using pre-existing resources. Resulting similarities between service functions show that the set of measures calculations can be combined to obtain more complex and significant information concerning functions inter-relations. This combination of measures can be conducted by implementing more similarity methods or by defining additional rules of inference. This process can be defined as Web services mining in the sense that helps to discover unknown relationships between functions. Inference is a key issue for maintainability and evolution of the ontology; inference rules generate new inter-relationships between functions and help to answer constrained queries regarding asserted inter-relationships in the ontology. Experimental results show that the overall process towards the automation of public Web services mining based on ontology population and structural similarity measures is feasible and can be completely automated.

The next steps of this research are the implementation and combination of more sophisticated similarity measures to facilitate automatic discovery and composition of Web services. In particular, behavior similarity measures, data type comparison measures, and linguistic patterns will be designed and applied to discover deeper semantic relations between public available Web services.

ACKNOWLEDGEMENTS

Authors thank the ICYTDF and UAM for their financial support granted for the presentation of this research.

REFERENCES

[1] http://www.w3.org/TR/wsdl, last visited 15.03.2013.

[2] http://uddi.org/pubs/uddi_v3.htm, last visited 15.03.2013.

[3] http://www.ebxml.org, last visited 15.03.2013.

[4] M. Rodríguez, M. Crasso, A. Zunino, and M. Campo, "Improving Web Service descriptions for effective service discovery," Science of Computer Programming, vol. 75, no. 11, Elsevier Science, 2010, pp. 1001-1021, doi: 10.1016/j.scico.2010.01.002.

[5] S. McIlarith, T. Cao Son, and H. Zeng, "Semantic Web Services," IEEE Intelligent Systems, vol. 16, no. 2, IEEE, 2001, pp. 46-53.

[6] http://www.w3.org/Submission/OWL-S, last visited 15.03.2013.

[7] http://www.wsmo.org, last visited 15.03.2013.

[8] http://www.w3.org/TR/sawsdl, last visited 15.03.2013.

[9] L. Hamel, M. Graiet, M. Kmimech, M. Bhiri, and W. Gaaloul, "Verifying Composite Service Transactional Behavior with EVENT-B," Proc. of the 5th European Conference on Software Architecture (ECSA 2011), Springer LNCS, September 2011, pp. 67-74, doi: 10.1007/978-3-642-23798-0.

[10] S. Chen, Z. Feng, H. Wang, and T. Wan, "Building the Semantic Relations-Based Web Services Registry through Services Mining," Proc. of the Eigth IEEE/ACIS International Conference on Computer and Information Science (ICIS 2009), IEEE, June 2009, pp. 736-743, doi: 10.1109/ICIS.2009.20.

[11] G. Zheng and A. Bouguettaya, "A. Service Mining on the Web," IEEE Transactions on Services Computing, vol. 2 no. 1, IEEE, January-March 2009, pp. 65-78, doi: 10.1109/TSC.2009.2.

[12] X. Zhang, Y. Ying, M. Zhang, and B. Zhang, "A Composite Web Services Discovery Technique Based on Community Mining," Proc. of the IEEE Asia-Pacific Services Computing Conference (IEEE APSCC), IEEE, December 2009, pp. 445-450, doi: 10.1109/APSCC.2009.5394087.

[13] L. Young-Ju and K. Chang-Su, "A Learning Ontology Method for RESTful Semantic Web Services," Proc. of the International Conference on Web Services (ICWS 2011), IEEE, July 2011, pp. 251-258, doi: 10.1109/ICWS.2011.59.

[14] A. Yousefipour, M. Mohsenzadeh, A. Ghari, and M. Sadegzadeh, "An Ontology-based Approach for Ranking Suggested Semantic Web Services," Proc. of the 6th International Conference on Advanced Information Management and Service (IMS 2010), IEEE, December 2010, pp. 17-22.

[15] Y. An, J. Geller, Y. Wu, and S. Ae Chun, "Automatic Generation of Ontology from the Deep Web," Proc. of the 18th International Workshop on Database and Expert Systems Applications (DEXA 2007), IEEE, September 2007, pp. 470-474, doi: 10.1109/DEXA.2007.43.

[16] M. Sabou and J. Pan, "Towards semantically enhanced Web service repositories," Journal of Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, no. 2, June 2007, pp. 142-150, doi: 10.1016/j.websem.2006.11.004.

[17] http://www.seekda.com, last visited 15.03.2013.

[18] T. Gruber, "A Transaltion approach to portable ontologies," Knowledge Acquisition, vol. 5, no. 2, June 1993, pp. 199-220, doi: 10.1006/knac.1993.1008.

[19] M. Bravo and M. Alvarado, "Similarity Measures for Substituting Web Services," International Journal of Web Services Research, vol. 7, no. 3, July 2010, pp. 1-29, doi: 10.4018/jwsr.2010070101.