LNBIP 62

Miguel-Angel Sicilia
Christian Kop
Fabio Sartori (Eds.)

Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science

4th International Workshop, ONTOSE 2010, held at CAiSE 2010
Hammamet, Tunisia, June 2010
Revised Selected Papers

Springer

Lecture Notes in Business Information Processing 62

Series Editors

Miguel-Angel Sicilia Christian Kop
Fabio Sartori (Eds.)

Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science

4th International Workshop, ONTOSE 2010
held at CAiSE 2010,
Hammamet, Tunisia, June 7-8, 2010
Revised Selected Papers

Springer

Volume Editors

Miguel-Angel Sicilia
University of Alcalá
Computer Science Department
Ctra. Barcelona km. 33.6, 28871 Alcalá de Henares, Madrid, Spain
E-mail: msicilia@uah.es

Christian Kop
Alpen-Adria-Universität Klagenfurt
Angewandte Informatik/Application Engineering
Universitätsstraße 65-67, 9020 Klagenfurt, Austria
E-mail: chris@ifit.uni-klu.ac.at

Fabio Sartori
University of Milan-Bicocca
Department of Computer Science, Systems and Communication (DISCo)
viale Sarca, 336, 20126 Milan, Italy
E-mail: sartori@disco.unimib.it

Library of Congress Control Number: 2010937278

ACM Computing Classification (1998): J.1, H.3.5, I.2.4, I.2.11

ISSN 1865-1348
ISBN-10 3-642-16495-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-16495-8 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180 5 4 3 2 1 0

Preface

As in the previous editions, ONTOSE 2010 once again was a discussion platform for all kinds of aspects on ontologies, service sciences, information systems and software engineering. The ONTOSE 2010 proceedings reflect this open discourse and hence contain contributions on various aspects of the above-mentioned research areas. A total of 25 contributions were received. These contributions underwent a full peer review with at least two referees, and finally 10 papers were presented at the Workshop and collected in these proceedings. The presentations particularly focused on enterprise and service architectures, ontology application, visualization and query expansion as well as on ontologies for services.

In the area of enterprise and service architecture the paper written by Thomas Kohlborn, Christian Luebeck, Axel Korthaus, Erwin Fielt, Michael Rosemann, Christoph Riedl and Helmut Krcmar outlines how relationships between services can be a basis for service bundling. The work of Remigijus Gustas describes an approach which better supports the integration of static and dynamic aspects in service architectures. The third paper by Sabine Buckl, Florian Matthes, Christopher Schulz and Christian M. Schweda addresses research questions concerning interrelating enterprise architecture (EA) (i.e., questions concerning the structure of the framework for interrelating EA, the types of relationship used, utilization of the framework for development and evolution of EA).

The three papers in the Ontology Application section are devoted to ontologies as formal descriptions for several applications. The first paper, written by Benjamin Diemert, Marie-Hélène Abel and Claude Moulin, discusses the usage of ontologies for digital information exchange. In the second paper, Jinsongdi Yu, Xia Wang and Peter Baumann present its usage for geospatial coverage information. Finally, Diego Bernini, Daniela Micucci and Francesco Tisato present their research work about an interoperability model based on spatial concepts, which can be used in software components in the ubiquitous computing domain.

Two papers were subsumed to the third section of these proceedings. The first one, written by Juan Garcia, Francisco Garcia and Roberto Theron, focuses on several strategies to view the coupling between ontology elements. In the second paper, Fabio Sartori and Matteo Palmonari describe query expansion and how ontologies can be used therein.

The last section, Ontology for Services, completes these proceedings and focuses once again on the two key topics of the workshop (i.e., ontology and services). Florie Bugeaud and Eddie Soulier present a formalization of service upstream in the telecommunication domain. Finally, Miguel-Angel Sicilia and Manuel Mora address the need to use the REA enterprise ontology.

The production of these proceedings, with all these interesting research results, was possible because many people supported the workshop. We would like to thank the authors for their valuable contributions, the reviewers for their substantial reviews and the CAiSE workshops organization chairs Jolita Ralyté and Pierluigi Plebani who supported the process.

June 2010

Miguel-Angel Sicilia
Christian Kop
Fabio Sartori

Organization

Workshop Co-chairs

Miguel-Angel Sicilia	University of Alcalá, Spain
Christian Kop	Alpen-Adria-Universität Klagenfurt, Austria
Fabio Sartori	University of Milano–Bicocca, Italy

Sponsoring Institution

Department of Computer Science, Systems and Communication (DISCo), University of Milano–Bicocca, Milano

Program Committee

Sudhir Agarwal	Universität Karlsruhe (TH), Germany
José Angel Olivas	Universidad de Castilla–La Mancha, Spain
Stefania Bandini	University of Milano–Bicocca, Italy
Djamal Benslimane	LIRIS, Université Claude Bernard, Lyon, France
Pierre Bourque	Engineering School of the Université du Québec, Canada
Pierre-Jean Charrel	Université Toulouse le Mirail, France
Jürgen Ebert	University of Koblenz, Germany
Werner Esswein	TU Dresden, Germany
Johann Eder	Alpen-Adria-Universität Klagenfurt, Austria
Sergio España Cubillo	Universidad Politecnica de Valencia, Spain
Raimund Feldmann	Fraunhofer Center for Experimental Software Engineering, Maryland, USA
Alexander Felfernig	Graz University of Technology, Austria
Anthony Finkelstein	University College London, UK
Gerhard Friedrich	Alpen-Adria-Universität Klagenfurt, Austria
Faïez Gargouri	l'Institut Supérieur d'Informatique et du Multimédia de Sfax, Tunisia
Olly Gotel	New York City, USA
Nicola Guarino	Laboratory of Applied Ontology, ISTC–CNR, Italy
Remigijus Gustas	Karlstad University, Sweden
Peter Haase	Universität Karlsruhe (TH), Germany
Wolfgang Hesse	Philipps–Universität Marburg, Germany

Paul Johannesson	Stockholm University, Sweden
Miltiadis Lytras	American College of Greece, Greece
Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria
John-Jules Charles Meyer	University of Utrecht, The Netherlands
Daniela Micucci	University of Milano–Bicocca, Italy
Roland Mittermeir	Alpen-Adria-Universität Klagenfurt, Austria
Jari Palomäki	Tampere University of Technology/Pori, Finland
Fernando Silva Parreiras	University of Koblenz, Germany
Daniel Rodríguez	University of Alcalá, Spain
Francisco Ruiz González	Universidad de Castilla–La Mancha, Spain
Christelle Scharff	Pace University New York, USA
Francesco Tisato	University of Milano–Bicocca, Italy
Markus Zanker	Alpen-Adria-Universität Klagenfurt, Austria

Table of Contents

Identification and Specification of Relationships as the Foundation for Service Bundling

Thomas Kohlborn[1], Christian Luebeck[2], Axel Korthaus[1], Erwin Fielt[1], Michael Rosemann[1], Christoph Riedl[2], and Helmut Krcmar[2]

[1] Faculty of Science and Technology, Queensland University of Technology, 126 Margaret Street, 4000 Brisbane, Australia
{t.kohlborn,axel.korthaus,e.fielt,m.rosemann}@qut.edu.au
[2] Lehrstuhl für Wirtschaftsinformatik, Technische Universität München
85748 Garching b. München, Germany
{luebeckc,riedlc,krcmar}@in.tum.de

Abstract. Service bundling can be regarded as an option for service providers to strengthen their competitive advantages, cope with dynamic market conditions and heterogeneous consumer demand. Despite these positive effects, actual guidance for the identification of service bundles and the act of bundling itself can be regarded as a gap. Previous research has resulted in a conceptualization of a service bundling method relying on a structured service description in order to fill this gap. This method addresses the reasoning about the suitability of services to be part of a bundle based on analyzing existing relationships between services captured by a description language. This paper extends the aforementioned research by presenting an initial set of empirically derived relationships between services in existing bundles that can subsequently be utilized to identify potential new bundles. Additionally, a gap analysis points out to what extent prominent ontologies and service description languages accommodate for the identified relationships.

Keywords: Service bundling, service description language and ontology.

1 Introduction

The creation of bundled offers of services and goods with distinguishing and superior characteristics compared to existing offers has long been recognized as an opportunity for companies to increase their competitive advantages over rival contenders in the market [1]. Generally, a bundle represents a package that contains at least two elements and presents a value-add to potential consumers. In the travel industry, for example, bundling is common and well-known. For instance, a travel business might offer a service bundle that combines three nights of accommodation in a hotel on a tropical island together with a one-day diving trip. Consumers might perceive the diving trip as a distinguishing characteristic of the offer and a decisive factor for selecting this bundle.

While a considerable amount of literature addressing the process of service design or new service development can be found today, much less is known about approaches

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 1–16, 2010.

that facilitate the creation of adequate service bundles. Despite the fact that companies across all industry sectors with increased market pressures are challenged by the issue of service bundling [2], only little guidance has been provided so far for the identification of potential bundle candidates and for the actual process of bundling. Previous research introduced a service bundling approach that can be positioned within the area of Semantic Science as it utilizes the description of a service to identify relationships to reason about the suitability of a potential new service bundle [3, 4].

In this paper, we aim to extend the aforementioned research by providing empirical insights into a first set of relationships and its coverage by existing service description languages and ontologies. Not only do we provide insights into foundational aspects and the process of bundling, we also extend the proposed service bundling method that will support organizations in finding service bundles that could potentially be offered to service consumers. The overall research can be positioned in the area of Design Science: by extending a service bundling method, the objective is to develop an "artifact" to "solve a contemporary problem" [5]. However, the evaluation of the artifact, although it is a constituent part of Design Science, is not in scope of this paper, since we report on research in progress.

The remainder of this paper is structured as follows. Based on the problem description that has been provided in this section, we will briefly introduce our service bundling method to provide the means to position the remainder of this paper appropriately. Subsequently, a first set of relationships derived from existing bundles is presented that can be utilized by our service bundling method to identify potential new bundles. Then, existing, prominent service description languages and ontologies are analyzed in regard to their coverage of the identified relationships. The paper ends with a conclusion and directions for further research.

2 Positioning Previous Research

The proposed method is targeted at the identification of possible service bundles by supporting the early stages of the bundle creation process. The method therefore focuses on limiting the solution space of possible bundles, using indicators that express some form of bundling motivation. It is important to point out that this method is not supposed to omit the evaluation of bundles by a domain expert. It has to be acknowledged that the domain expert is still needed to evaluate the overall feasibility of bundles, since this requires complex analysis, often utilizing tacit knowledge across a range of different disciplines (e.g. economy, marketing, legal). Rather, the aim of this method is to limit the scope of the necessary evaluation for the domain expert. This is in particular relevant with a large number of services and, therefore, many bundling options. The proposed approach leverages existing service descriptions and does not necessitate a time-consuming step of (manually) explicating relationships between services as it is the case with the method described by Baida [6], for example. Instead, commonalities of attributes indicate such a relationship. As long as services are consistently described and attributes relevant for this bundling approach are present, the proposed method can be employed. Moreover, Baida [6] relies on a given customer demand to drive the creation of service bundles. While useful for situations where customer demand is well known and understood, poor performance can be expected from this approach when demand is hard to capture or anticipate. Furthermore, the economically desirable situation where customer demand is induced by a new service offering is not supported at all. Our proposed method explicitly targets the latter case by focusing on the creation of new and innovative service bundles. Therefore, customer demand is not utilized to reason about the suitability of potential bundles in this method. Instead, the driving source of this method is a repository of services that are available for bundling. Depending on the given context, this repository might consist of the services of a single provider, a provider network or even contain all available services in a service ecosystem.

Herrmann et al. [7] found that functionally complementary components in a bundle lead to high intentions to purchase compared to bundles in which no complementary components are present. The authors state that, "*as the relationship among the components increased from 'not at all related' through 'somewhat related' to 'very related', intention to purchase also increased*". The proposed method builds upon these findings and the conjecture that other commonalities or relationships between services can also indicate potentially useful bundles. We define the term relationship as *a connection, whose existence can be evaluated by a logic expression* utilizing service description attributes. Every relationship refers to previously specified attributes (e.g. location of the hotel, destination of the flight) and evaluates them using a given logic (e.g. distance between destination airport and location of the hotel). This evaluation can be realized ranging from simple value comparisons of single attributes to complex algorithms using multiple attributes.

We distinguish between two types of relationships, namely *generic relationships* and *specific relationships*. A generic relationship is used independently of a concrete domain and is applied on *possible bundles*, which comprise any combination of two or more different services to one package. These relationships evaluate connections of a general nature that can be found across a range of different domains. The evaluation of generic relationships does not require a domain-specific awareness and will lead to *generic bundles*. A specific relationship only applies to certain domains and can be tailored for concrete bundling scenarios. As visualized in Fig. 1, applying generic and specific relationships leads to the derivation of potential *specific bundles*. As pointed out previously, those bundles have to be evaluated by a domain expert to check the feasibility of the proposed bundle. Therefore, *domain knowledge* is utilized to get from specific bundles to *feasible bundles*.

As this method represents a structured, explicable approach, it is possible to develop a tool to support the identification of service bundles. The tool requires an explicit service description language, clearly defined relationships and corresponding

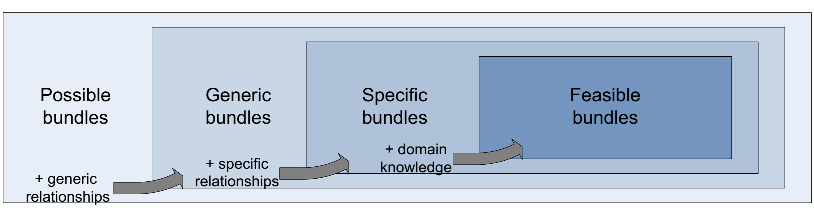


Fig. 1. Constraining the Solution Space to Derive Service Bundles

logic as well as consistently described services. A tool could evaluate chosen relationships for all possible service bundle combinations, therefore creating a short-list of feasible bundles.

3 Empirical Derivation of an Initial Set of Generic Relationships

3.1 Analysis of Existing Service Bundles

The previous section has introduced the notion of relationships in a bundling context and explained the difference between generic and specific relationships. This section aims at deriving a first working set of generic relationships.

An empirical approach was chosen to analyse existing service bundles and to identify relationships that exist between the components of the bundle. For this purpose, a survey in form of a questionnaire was designed listing a total of 28 bundles. The following description of the empirical derivation generic relationships in inspired by [8].

The objective of this questionnaire was to collect a set of generic relationships among components a part of bundles as perceived by ordinary people. For this purpose we provided a list of 28 bundles. These bundles were either concrete existing real-world bundles (e.g. Flight Brisbane to Frankfurt + Accommodation in Hotel Savigny**** Frankfurt City) or generic representations for common combinations (e.g. Phone Landline + ADSL). Since it is possible that relationships between goods might be identified that are also valid for services, the questionnaire has not been limited to services. Instead it contains bundles that are composed of services, goods or a combination of both. For each bundle the participant had to identify which relationship among the components probably influenced most the designer of the bundle. The survey took place at the end of 2009, after ensuring that appropriate resources would be available to administer the survey and analyse the results. The survey was sent out as an attachment to an email, so that responses were either in the form of filled-out print-outs of the survey or electric scans of those filled-out versions. After an initial analysis of the first version of the survey, which included proof reading and discussing the potential understanding of certain bundles by participants, we decided to provide an initial example. For the first bundle of the questionnaire *Flight Brisbane to Frankfurt + Hotel Savigny**** Frankfurt City* the example was given that both bundle components have a common location, since the arrival city of the flight is the same as the location of the hotel. The questionnaire was completed by five people from different backgrounds based on convenience sampling.

3.2 General Findings

Since the questionnaire was highly explorative, the results varied strongly in terms of quality and creativity. Some participants were biased by the given example and evaluated the bundles mostly with regard to the relationships already provided by the example. Some participants focused on customer demand. While it is obvious that most of the bundles relate to a certain customer demand, this information is not available in a service repository. Positioned as a bottom-up approach, this method does not require a previously defined customer demand and findings related to customer demand were disregarded.

3.3 Identified Generic Relationships

Location: Apart from the given example, all participants identified other bundles with location relationships. This was mainly found in service bundles that included a transportation service. In one instance a location relationship was also identified for a service bundle that required the customer's physical presence for service invocation and consumption. One participant also noted varying degrees of the relationship stating that within one bundle the locations are 120km apart.

Time: Two participants indicated that relationships in terms of timing are also present. The mentioned bundles contain services that are consumed in a certain order. Therefore, along the lines of the location relationship, the time relationship indicates temporal availability, a requirement for the sequential consumption of bundle components.

Resource: Three participants found relationships regarding the resources of a bundle. For example in the bundle *Phone Landline + ADSL* it was noted that both components share the same transport medium. For the bundle *Phone Landline + Unlimited Local & Nation Calls* and the bundle *Wii Gaming Console + Game Wii Sports* one person noted that the first component is a required resource for the second component. Along these lines, two participants identified that all products in the Bundle *Multi Purpose Cleaner, Windscreen Wash, Wash & Wax, Paint Protector, Glass Wipes, Bucket, Sponge* target the same object, a customer's car. These findings are all based on the fact that services can require external resources. Hence this relationship is called Resource relationship.

Event: Two participants found that the components in some bundles support certain Events. In the bundle *Cup of Coffee + Cake of the Day* the events *Break* or *Snack* were seen, while the bundle *2 Pizzas + 2 Movie Tickets* might apply for the events *Night with friends* or *Going out*. The notion of an event can be seen as a generic type of customer demand, since a general context is applied.

Customer group: Two participants also identified bundles with components targeting a certain customer segment or group. For example the bundle Wii *Gaming Console + Game Wii Sports* was classified as targeting *gamers*. One participant indicated that most bundles target either a business or leisure customer segment.

Compensation: Two participants stated that some bundles contain components to counterbalance disadvantages of another component. This relationship was seen in the bundle *Flight Brisbane to Frankfurt + Single-Trip Essentials-Travel Insurance Worldwide*, where the insurance is a means to compensate for the risk of international travel. Within the bundle *Flight Brisbane to Frankfurt + Expedia Global Calling Card*, the calling card was seen as a means to ease communication challenges imposed by the long distance and within the bundle *Flight Brisbane to Frankfurt + Fly-Green with Terrapass (Carbon Offset Program)*, the carbon offset program was seen as a countermeasure to the increased personal carbon footprint due to the flight.

Capability: Two participants also found bundles with components related to their purpose, usage or capability. For example, the components in the bundle *Multi Purpose Cleaner, Windscreen Wash, Wash & Wax, Paint Protector, Glass Wipes, Bucket, Sponge* were all identified as serving the purpose of cleaning a car.

Complementarity: Four participants also indicated some form of general complementarity relationship between the components.

Compatibility: One participant evaluated the bundles in terms of the compatibility of the components. He stated that within multiple bundles the components are technically compatible.

Customer demand: One participant indicated that nearly all bundles have related components, as they satisfy a certain customer demand.

Category: Two participants also indicated that most bundle components reside in the same or a similar category.

The set of generic relationships derived from existing service bundles will have to be supplemented by domain-specific relationships. The identification of the latter type of relationships, however, is beyond the scope of this paper and remains subject to future work.

4 Detailing the Identified Relationships

The previous section has identified relationships between bundle components based on real-world services. This section will discuss the usability of these findings for a first working set of generic relationships. This first set does not aim to be complete. Rather its purpose is to serve as a starting point for the development of the prototype. It has to be seen as a creative artifact of design science, which can evolve over time. Furthermore, it should be noted that a certain degree of interpretation was applied to categorize the findings.

The relationships *location* and *time* have been identified by multiple participants. It seems logical that the spatial and temporal availability of services plays an important role for many service bundles. This is especially relevant for services provided by brick-and-mortar providers like restaurants, cinemas or transportation providers. Both relationships will therefore be included in this first working set. The *resource* relationship has also been mentioned multiple times. Since bundling services, that share certain resources, might lead to efficiencies on service provider and consumer side, this relationship will be included in this first set. The relationships *event* and *customer* have also been discovered by multiple participants. Both relationships are regarded as relevant for bundling and will therefore also be included. The relationships *compensation*, *complementarity* and *compatibility* have distinct characteristics. The evaluation of these relationships is complex and very domain specific. For example, it seems unrealistic to create an explicit logic for the complementarity of two services solely based on their service description, since complementarity can be based on multiple domain dependent factors. These relationships therefore have to be evaluated by a domain expert. Once a domain expert has identified explicit relationships between services, this information can be fed into the service description. This working set will therefore cover all explicit connections between services as a *linkage* relationship. The relationships *category* and *capability* will not be included in this set. Although the questionnaire results indicated promising results, it would be necessary to create a taxonomy of categories and capabilities before services could be sufficiently described. The creation of classification schemes for these relationships would go beyond the scope of this paper and should therefore be considered future work. Lastly, the relationship *customer demand* was identified. Since this method is a bottom-up approach, which explicitly avoids customer demand for multiple reasons as already outlined in section 2, this relationship will not be included in the set.

Our first working set of relationship therefore includes a total of six relationships, i.e. the location, time, resource, event, customer and linkage relationship. To utilize these relationships in a prototype, attributes for the service description have to be specified. Based on these attributes, logic has to be developed that is capable of evaluating the relationships. The following paragraphs will cover each relationship in detail. An exemplary model for appropriate attributes and the associated logic is given.

Location

A location relationship exists between two services, if both services are available at the same location or within a previously specified range. The service description needs to include the spatial availability of a service to evaluate this relationship. This has to be described by a list of locations, which can be either political or geographical. Political locations can be captured on different detail levels, namely continent, country, state, city, street, building and unit. A location implicitly covers all detail levels that are not described. If, for example a service is available everywhere within one city, the attributes for street, building and unit are left empty. A geographic location can be described as a coordinate specifying longitude and latitude. These details should be provided using WGS84, a reference system used by the Global Positioning System (GPS). The location can also be described as a shape using a minimum of 3 coordinates. Using shapes, any spatial area can be almost perfectly described. It is therefore possible to describe single points of service availability as well as complex areas, politically as well as geographically. A UML-Model visualizing the described attributes is shown in Fig. 2.

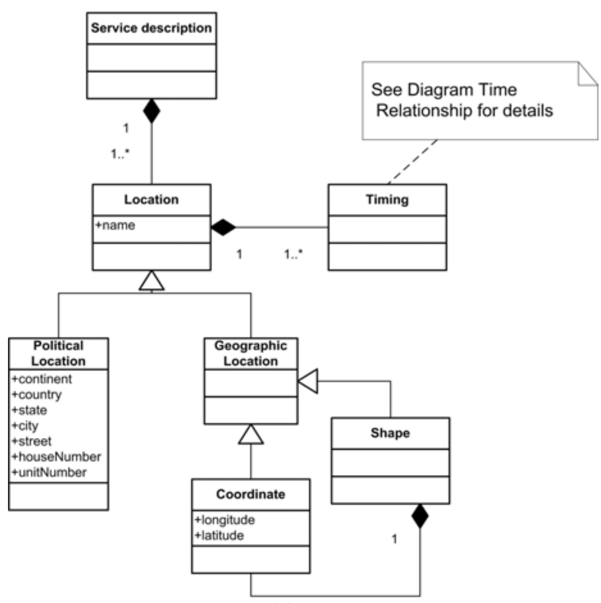


Fig. 2. UML-Model of Attributes for the Location Relationship

The logic evaluates the distance between any two given services. If services have multiple locations, the closest distance is used. If political locations are used, they will be geo-coded, so that calculations on distances are possible. If the spatial availability is described by shapes, the closed distance from any point in the shape is used. The calculated distances between the services are compared with the maximum allowed distance, which has to be set as a parameter of the relationship. To allow for varying degrees of strength, the logic scores the relationship on a scale from 0 to 100, 100 meaning both services have the same location and 0 meaning the distance between the services locations is greater than the specified maximum distance.

Time

Along the lines of spatial availability, the time relationship covers the temporal availability of a service. A time relationship between two services exists when both services can be consumed at the same time or within a specified time frame. The service description needs to include a list of all time periods in which the service is available. These periods can be specified on two levels for all specified locations and separately for each location. To specify the timing for one specific location, time periods can be added to the location-element as visualized in Fig. 2. A time period consists of the start and end time of the day. A recurrence pattern can also be specified to easily describe multiple instances. The recurrence can be limited using a start and end date. In addition to the availability of the service, the typical duration of service consumption also has to be specified. This is required to evaluate sequences of services. A model of the relevant attributes for this relationship, including the possible recurrence pattern, is shown in Fig. 3.

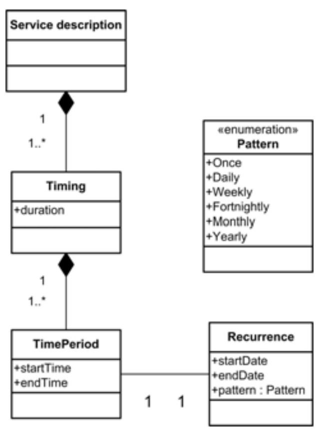


Fig. 3. UML-Model of Attributes for the Time Relationship

The logic for evaluating a time relationship can operate in two distinct modes, namely parallel and sequential mode. In parallel mode, the logic evaluates whether services are available at the same time, meaning there is an overlap in temporal availability. The strength of the relationship depends on how big the overlap is. The logic scores 100 if both services are available at the same times and 0 if both services are never available at the same time. In sequential mode the logic evaluates whether the services can be arranged as a sequence. In this mode a maximum time span between the service consumption has to be specified. The logic then tries to accommodate both services in a sequential order, taking into account this time span and the specified durations of service consumption. The highest score of 100 is reached if both services can be brought into a sequence without a gap in between. The greater the gap, the lower the score is. If the gap is greater than the specified maximum time span the specified maximum 0 is scored.

Resource

Services may also be related because they share the same resources. Resources can play different roles for a service. In this model a resource might be used as the input, output or the target object of a service. Resources can also vary in their importance for a service. For example, resources, which the customer has to bring into the process, might be of higher importance.

The service description has to include a list of all used resources. For every resource, the type of usage (input, output or target) and the importance in form of a weight have to be specified. The weight provides a generic way of expressing the relevance of a resource for a given service. The attributes are visualized in Fig. 4.

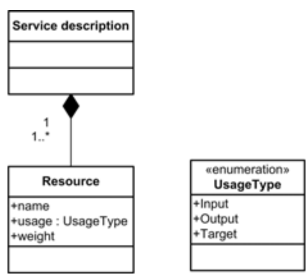


Fig. 4. UML-Model of Attributes for the Resource Relationship

The logic for the resource relationship can operate in different modes. It can be configured to evaluate the used resources regardless of the type of usage (input, output or target object) or specifically target only one specific usage type. It can also be configured to identify resources with different usages (for example a resource is output of one service and acts as input for another service). The different operating modes can be combined in any order to support complex requirements. Independent from the chosen mode, the strength of the relationship is always scored based on the relation of matched resources to total resources used by both services. The more resources that are used by both services, the higher the score will be. A score of 100 denotes a perfect match. The weights assigned to the resources are used to weigh the overall score according to the importance of the resources.

Customer

Services can be interrelated because they are consumed by the same or similar type of customer. To evaluate this relationship the properties of the targeted customers have to be described.

The service description has to include a list of relevant attributes that describe the properties of targeted customers. These attributes can be of a generic nature (e.g. age, gender, income) or they can be domain-specific (e.g. movie preferences, willingness to take financial risks). In the current model an attribute can be specified either as a value range (e.g. age, income) or as a simple value (gender, movie category). Apart from attributes, the problems and goals that are targeted by a service can also be described. Attributes necessary for the customer relationship are visualized in the UML-Model in Fig. 5.

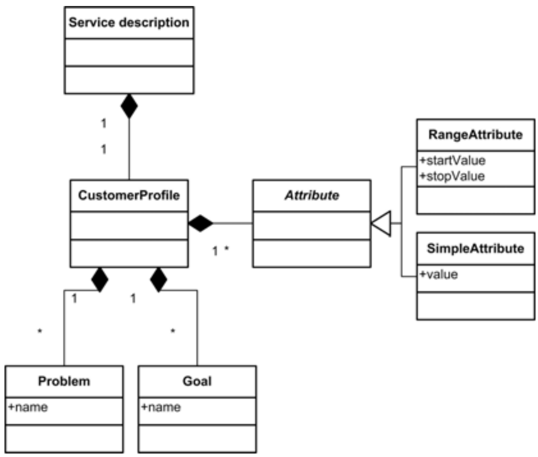


Fig. 5. UML-Model of Attributes for the Customer Relationship

The customer relationship logic evaluates the listed problems and goals and attributes of both services and scores how many of these can be found in both services. Value range attributes contribute towards the score in proportion to the overlap between the compared ranges.

Linkage

The linkage relationship is special and has distinct characteristics compared to the former relationship types. This relationship covers all ex ante and explicitly specified relationships between two services. Two types of relationships are relevant in this context, positive and negative relationships. Positive relationships might be enhancing, complementing, compatible or compensating. These relationships indicate bundling opportunities. Negative relationships might be substituting or excluding/ incompatible. These relationships discourage bundling.

The service description has to include links between related services specifying the types of relationships. Fig. 6 visualizes this relationship.

To evaluate the linkage relationship, the score value of each relationship has to be specified beforehand. The logic then assigns the applicable score between 0 and 100 to each service relationship (e.g. a service enhancement might be valued at 100 points, while a complementing link might be worth 70 points).

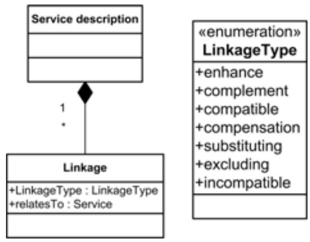


Fig. 6. UML-Model of Attributes for the Linkage Relationship

Event

The event relationship can be used to identify services that are used for specific occurrences. Events can be defined for any kind of happening, incident or occurrence (e.g. bushfire, merger and acquisition, Christmas).

For the event relationship the service description has to include a list of events that have relevance for the service. This is visualized in Fig. 7.

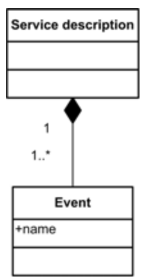


Fig. 7. UML-Model of Attributes for the Event Relationship

The logic for the event relationship evaluates the count of events that apply to both services. The relation of matched events to total events determined the score, ranging between 0 and 100. A score of 100 means that both customer profiles are identical and 0 means that no similarities in the profiles exist.

5 Analyzing Service Description Languages and Ontologies

5.1 Contrasting Description Languages and Ontologies

Description languages and ontologies are used as a structured approach to describe services. Ontologies are used to define high level concepts and the relationship between these concepts and therefore can be used for service discovery and analysis as well as enabling reasoning about services. Ontologies can be created for multiple purposes with varying characteristics. One common scenario is to define a taxonomic structure of terms associated with their definitions, so-called lightweight ontologies. On the other hand, foundational ontologies such as the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) use Axioms in order to remove terminological and conceptual ambiguities and to facilitate mutual understanding and inter-operability among people and machines [9]. Ontologies can often be extended, refined or inter-connected using mediators [10].

Efforts that do not employ a formally described ontology are commonly referred to as description languages. These efforts often use modeling languages such as UML or XML, or describe their concepts in an informal fashion. Multiple efforts to describe services can currently be found. Some ontology based efforts aim at extending the ideas of the semantic web to services, mainly to enable machine-based agents to semantically discover services and to reason about integrational aspects [11]. Other ontologies and description languages try to facilitate topics such as service bundling or target a business-oriented view.

Since the evaluating logic of a relationship is based on the service description, it is desirable to have an overview of the available options to describe a service as well as whether the necessary attributes for the previously introduced generic set of relationships are supported. Therefore, prominent efforts will be described and analyzed regarding how these efforts are generally suited for the bundling identification method. Following this, all efforts will be evaluated regarding their support for the needed attributes.

5.2 Analyzing Description Languages and Ontologies

WSMO. The Web Service Modeling Ontology (WSMO) [10] is based on the Web Service Modeling Framework (WSMF) [12], which it extends. It is comprised of 4 main areas: Ontologies, providing the terminology; goals that are reached using a service; Web service descriptions, defining various aspects of a service, and mediators, bridging the gap between different terminologies. WSMO acts as meta-model layer and does not go into great detail. For non-functional properties WSMO references the Dublin Core, an organization providing interoperable metadata standards that support a broad range of purposes and business models [13]. Using the Dublin Core concept *dc:coverage*, WSMO can be used to add information about the temporal and spatial availability of a service. The duration of a service consumption can be specified using *dc:format*. The concept of a relation can be utilized to express linkages between services. WSMO capability to import or mediate between other ontologies can be used to define the necessary concepts for the remaining attributes.

OWL-S. The Semantic Ontology Web Language (OWL-S) [14] uses a similar approach to WSMO. Based on the Ontology Web Language (OWL), OWL-S provides a framework to describe capabilities and properties of a web service. OWL-S aims at supporting a high level of automation throughout the complete lifecycle of a service. It is structured in 3 main parts: The service profile describing capabilities and functions used for advertising and discovery; the process model gives a detailed description of a services methods and the grounding providing technical details on how to access the service. OWL-S has a highly technical view and is mostly concerned with the automated discovery of web services by matching input, output, preconditions and results of services. Only the notion of resources in the form of input and output is present in OWL-S. With OWL-S being based on OWL, providing high-level semantic concepts, all necessary attributes could be implemented by deriving the concepts of OWL.

USDL. The Universal Service-Semantics Description Language (USDL) is very similar to OWL-S, but is based on a universal ontology using WordNet, a lexical database semantically linking related words/concepts [15], instead of having domain-specific ontologies and therefore avoiding the semantic aliasing problem [16]. USDL defines high level concepts to semantically annotate WSDL-Documents and also aims for automated service discovery and composition. None of the required attributes are explicitly described in USDL, but through utilizing WordNet it is possible to describe all necessary attributes. That being said, it has to be acknowledged that the evaluation of concepts described through WordNet might prove difficult, since the semantic richness of a lexical approach offers various ways of expressing one fact. USDL is therefore categorized as not describing any of the required attributes.

WSDL-S. The Semantic Web Service Description Language (WSDL-S) by Akkiraju et al. [17] uses XML to enrich WSDL with semantics. The authors describe how to use the extensibility elements of WSDL to create semantic annotations. WSDL-S is a very high level of abstraction and aims purely at web services. Rather than an actual description language it provides a method to combine XML-based WSDL descriptions and ontology based OWL descriptions. Hence, no description on actual attributes is present.

Oberle et al. [11]. The service ontology described by Oberle et al. [11] uses a modular approach. It is based on the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [9] as the foundation. This foundation can be extended by so-called core modules, providing semantics for various areas including functional and legal aspects, core service descriptions, and business models. The ontology can then be further refined using industry modules. This ontology aims at services in the internet of services. It is high level and does not go into the details of concrete modules, as the focus of this work is the modular use and inter-linkage of ontologies. No description for the required attributes is present in this work, but all attributes can be implemented as a new module based on DOLCE.

OBELIX. The OBELIX (Ontology-Based Electronic Integration of Complex Products and Value Chains) project [2] developed a service ontology to describe service bundles. It is component-based and allows a description of services and their relationship to each other. The ontology offers three top-level viewpoints, service value, service offering and service process. It defines a concept to describe the functions of individual services within a bundle. The project aims at describing real-world services making this ontology particularly relevant for this review. Within the service offering viewpoint, the concept of service bundles comprised of other service bundles or elementary services is defined. Furthermore services or service bundles can be linked to express a certain dependency (i.e. enhancing, supporting, bundle, substitute, excluding, optional bundle). The notion of a customer profile is also present. The service value viewpoint describes the customer perspective by a customer's demands and sacrifice. The sacrifice (what the customer is willing to give up for receiving the service) can be monetary or some kinds of relationship costs. OBELIX is developed to support the configuration of service bundles using customer demand and explicated service dependencies. It therefore does not explicitly provide any extension mechanism, but it is possible to extend the ontology itself.

Ferrario & Guarino [18]. The ontology by Ferrario and Guarino [18] tries to model a service as a layered set of interrelated events. It identifies five different layers, i.e. service value exchange, service process, service acquisition, service bundling and presentation, and service commitment. The ontology is business oriented and tries to provide a framework that fits real world services. This effort has to be considered as early work and does not go into much detail on specific layers. A concrete representation of a description is missing and none of the necessary attributes are described.

Colombo et al. [19]. The description language by Colombo et al. [19] uses UML as the foundation for modeling. The author aims at providing a conceptual model describing actors, activities and entities involved in a Service Oriented Architecture. In contrast to WSMO or OWL-S, this work tries to provide information for human-readers. The model specifically states the entity Service Description but leaves specifying the details of the entity to the user of the model. It is therefore regarded as having a high level of abstraction. It partially supports the notion of resources, as they can be used as the output of a service. Services can also be linked to each other, but only in the form of a composition. The model does not provide any extension mechanisms.

O'Sullivan [20]. The effort by O'Sullivan [20] focuses on non functional properties of a service. He addresses the research question of what would be a domain independent taxonomy that is capable of representing the non-functional properties of conventional, electronic and web services. He analyzed services from various domains and created 80 conceptual models in 10 different categories. The author aims at describing each category in as much details as possible to cover any kind of existing service characteristic in terms of non function properties. Therefore this work exposes a high level of details. The work includes all required concepts to model the spatial and temporal availability required for the relationships. The remaining attributes concerning resources, events, customer profiling and linkage are not described.

5.3 Comparative Analysis

A comparative overview regarding the performance of each description language or ontology in respect to its coverage of identified relationships and specified attributes is visualized in Table 1.

Table 1. Gap Analysis of Service Ontologies and Description Languages

Approach	Location			Time			Resource		Event	Customer			Linkage
	Political	Geo-graphical	Geo. Shapes	Periods	Duration	Recurrence	Type	Weight		Problems	Goals	Attributes	
WSMO [8]	S	S	S	S	S	-	-	-	-	-	P	-	S
OWL-S [12]	-	-	-	-	-	-	P	-	-	-	-	-	-
USDL [13]	-	-	-	-	-	-	-	-	-	-	-	-	-
WSDL-S [15]	-	-	-	-	-	-	-	-	-	-	-	-	-
Oberle et al. [9]	-	-	-	-	-	-	-	-	-	-	-	-	-
OBELIX [2]	-	-	-	-	-	-	P	-	-	P	P	-	S
Ferrario Guarino [16]	-	-	-	-	-	-	-	-	-	-	-	-	-
Colombo et al. [17]	-	-	-	-	-	-	P	-	-	-	-	-	P
O'Sullivan [18]	S	S	S	S	S	S	-	-	-	-	-	-	-

(S=Supported, P=Partially Supported/Similar Concept, - =Not Supported/Not Described)

Summarizing the findings in the previous section, it has to be said that no ontology or description language currently implements all required attributes or employs similar concepts. Most ontology approaches aim at providing a semantic foundation to enrich the technical descriptions of web services, without going into detail on concrete attributes. Furthermore, most approaches can be extended or enriched to include the required attributes.

As indicated in Table 1, WSMO, already employing concepts for spatial and temporal availability, seems to be a good starting point, although it has to be acknowledged that profound knowledge in the area of ontologies is required to correctly implement these extensions.

6 Conclusion

This paper describes the logical extension of previous research related to defining a novel approach to identifying service bundle candidates. Because of its potential to combine innovation with cost-effective re-use of existing services, we envision that service bundling will become as important as new service development as, for example, can be seen in the growing attention for mash-ups. However, while the process of new service development has been extensively researched and conceptualized, the process of finding suitable service bundling candidates is still ill-defined. The proposed method is a contribution to Design Science research in the field of Information Systems. It represents an innovative artifact that extends the academic knowledge base related to service management.

The method utilizes service description languages as they provide the means to analyze relationships between services to identify potential new bundles. In this paper we presented an empirically derived set of initial relationships that can be used to derive generic bundles. Each identified relationship has been described in detail and specified regarding associated attributes. Finally, relationships and attributes have built the foundation to conduct a gap analysis involving prominent service description languages and ontologies. Findings indicate that none of the analyzed languages and ontologies provides sufficient scope and depth to cover the identified relationships and attributes.

Thus, research in the area of service descriptions has to be conducted to develop a universal service description language that is applicable across industries and covers business as well as software services or extend existing approaches such as WSMO. While we have shown that generic relationships can be derived from existing service bundles, it also remains further work to validate the general utility of these relationship constructs and to derive specific relationships for different domains. At this stage, the proposed relationships have to be seen as a working set, which will evolve as additional studies and evaluations are carried out. Further research should also be conducted in the area of analyzing the suitability of different bundling approaches in different application scenarios and their utilization of existing service description languages.

Acknowledgements. This research was carried out as part of the activities of, and funded by, the Smart Services Cooperative Research Centre (CRC) through the Australian Government's CRC Programme (Department of Innovation, Industry, Science and Research).

References

1. Lawless, M.W.: Commodity Bundling for Competitive Advantage: Strategic Implications. Journal of Management Studies 28, 267–280 (1991)
2. Akkermans, H., Baida, Z., Gordijn, J., Peiia, N., Altuna, A., Laresgoiti, I.: Value Webs: Using Ontologies to Bundle Real-World Services. IEEE Int. Systems 19, 57–66 (2004)
3. Kohlborn, T., Luebeck, C., Korthaus, A., Fielt, E., Rosemann, M., Riedl, C., Krcmar, H.: Conceptualizing a Bottom-up Approach to Service Bundling. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 129–134. Springer, Heidelberg (2010)
4. Kohlborn, T., Luebeck, C., Korthaus, A., Fielt, E., Rosemann, M., Riedl, C., Krcmar, H.: How Relationships Can Be Utilized for Service Bundling. In: Proceedings of the Americas Conference on Information Systems (AMCIS), Lima, Peru (2010)
5. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
6. Baida, Z.: Software-Aided Service Bundling. Intelligent Methods & Tools for Graphical Service Modeling. In: Dutch Graduate School for Information and Knowledge Systems. Vrije Universiteit, Amsterdam (2006)
7. Herrmann, A., Huber, F., Coulter, R.H.: Product and Service Bundling Decisions and Their Effects on Purchase Intention. In: Fuerderer, R., Herrmann, A., Wuebker, G. (eds.) Optimal Bundling: Marketing Strategies for Improving Economic Performance, pp. 253–268. Springer, Heidelberg (1999)
8. Pfleeger, S.L., Kitchenham, B.A.: Principles of Survey Research: Part 1: Turning Lemons into Lemonade. SIGSOFT Softw. Eng. Notes 26, 16–18 (2001)
9. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Wonderweb Deliverable D18. Ontology Library (Final) (2003), `http://www.loa-cnr.it/Papers/D18.pdf`
10. Roman, D., Lausen, H., Keller, U.: D2v1.3. Web Service Modeling Ontology (WSMO): WSMO Final Draft (2006), `http://www.wsmo.org/TR/d2/v1.3/`
11. Oberle, D., Bhatti, N., Brockmans, S.: Countering Service Information Challenges in the Internet of Services. Business & Information Systems Engineering 1, 370–390 (2009)
12. Fensel, D., Bussler, C.: The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications 1, 113–137 (2002)
13. Dublin Core Metadata Initiative Ltd.: Dublin Core Metadata Initiative (2010)
14. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: Owl-S: Semantic Markup for Web Services (2004), `http://www.w3.org/Submission/OWL-S`
15. Fellbaum, C.: Wordnet: An Electronic Lexical Database. MIT press, Cambridge (1998)
16. Bansal, A., Kona, S., Simon, L., Hite, T.D.: A Universal Service-Semantics Description Language. In: Third European Conference on Web Services, pp. 214–225. IEEE Computer Society, Los Alamitos (2005)
17. Akkiraju, R., Farrell, J., Miller, J.A., Nagarajan, M., Sheth, A., Verma, K.: Web Service Semantics - WSDL-S (2005), `http://lsdis.cs.uga.edu/projects/METEOR-S/WSDL-S`
18. Ferrario, R., Guarino, N.: Towards an Ontological Foundation For Services Science. In: Domingue, J., Fensel, D., Traverso, P. (eds.) FIS 2008. LNCS, vol. 5468, pp. 152–169. Springer, Heidelberg (2009)
19. Colombo, M., Di Nitto, E., Di Penta, M., Distante, D., Zuccala, M.: Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 48–60. Springer, Heidelberg (2005)
20. O'Sullivan, J.: Towards a Precise Understanding of Service Properties. Queensland University of Technology (2006)

Conceptual Modeling and Integration of Static and Dynamic Aspects of Service Architectures

Remigijus Gustas

Department of Information Systems, Karlstad University, Sweden
Remigijus.Gustas@kau.se

Abstract. The concept of service is comprehensible in various domains and it can be used for integrated modeling of business process and business data. Service-oriented modeling approach, which is presented in this paper, is based on a single type of diagram. It helps to reason about semantic integrity of different modeling dimensions across organizational and technical system boundaries. A well-known case study is analyzed to demonstrate how service interactions among enterprise organizational components can be represented for detecting the essential similarities and differences of business processes. The ultimate goal of this paper is to overview deficiencies of conventional modeling approaches and to present generic principles for computation neutral modeling of service architectures. Conceptual models of service interactions between organizational and technical components enable separation of crosscutting concerns in enterprise engineering without requirement to specify a complete solution. However, most information system methodologies are projecting the structural, interactive and behavioral aspects into totally different modeling dimensions. If service architectures are represented as loose collection of models, it is difficult to detect integrity problems of various diagrams.

Keywords: Service interactions, value flows, decomposition and generalization, static and dynamic aspects of service architecture.

1 Introduction

Enterprise business processes can be defined on different modeling levels of granularity and in various projections, which are typically defined by using totally different types of diagrams. It is common to the industrial versions of system analysis and design methods to distinguish disparate views and dimensions of enterprise architecture [28]. The Zachman Framework can be viewed as taxonomy for understanding different types of diagrams that are used for representation of organizational and technical system architecture. It defines separate dimensions of business application and data architecture such as Why, What, How, Who, Where and When. Detection of inconsistency and incompleteness among different architecture views and dimensions is a fundamental problem in most information system (IS) methodologies. This is one obvious reason of integrity problems in information system specifications. For instance, semiformal IS modeling methods such as structured analysis and design [11], [4] , [27] suffer from paradigm mismatch in the diagrams, which analyze business processes and business data in isolation.

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 17–32, 2010.

Unified Modeling Language (UML) [25] was developed with the ultimate goal for unifying the best features of the graphical modeling languages and creating a de facto industry standard for system development. However, semantic integration principles of different UML diagram types are not totally clear. UML notation has several some accidental, essential and fundamental weaknesses [13]. The fundamental deficiencies can be summarized as follows:

- Value flows between actors and service interactions cannot be explicitly captured in UML,
- It is unclear how to combine interactive, structural and behavioral aspects together in a single view.

Data flow modeling was the strength of structured analysis and design methods [11], [27]. UML also supports various types of associations between classes, actors and use cases, or between objects such as software or hardware components. However, it is not suitable for modeling direct associations between actors that define actor interaction flows outside technical system boundary. Modeling of service or data flows between organizational enterprise subsystems is awkward in UML. It is not clear how rich context of actor interactions can be expressed. If actor interaction flows cannot be explicitly captured, then they are viewed as tacit knowledge, which is hidden from enterprise architects. In this case, such important relations cannot be maintained by CASE tools.

One of the benefits of enterprise modeling is to analyze business processes for reaching consensus on how and by whom the processes are carried out. Most conceptual modeling methods that are intended for business process modeling are not using explicitly the concept of value flow. Value models, which include resource exchange activities among actors, can be viewed as design guidance in enterprise reengineering [21]. Declarative nature of value flows is very useful from system analysis point of view for one reason that flows have very little to do with the dependencies between business activities. The particular strength of value flows is separation of crosscutting concerns among organizational and technical subsystems. Each value flow between service requester and service provider can be further refined in terms of more specific interaction dependencies among organizational components. Service flows are suitable for discussing new configurations of processes with business developers, enterprise architects, system designers and users. Typically, business stakeholders distinguish among the past, current and future structure of processes that can be expressed on different levels of abstraction. Typically, business process modeling is not dealing with the notion of value flow, which demonstrates value exchange among actors involved in business processes [18]. Most information system methodologies are quite weak in representing the alternative value flow exchange scenarios and consequences if commitments are broken.

The treatment of fundamental deficiencies would require modification of the UML foundation. Introducing fundamental changes in UML constructs with the purpose of semantic integration of collection of models is a complex research activity. However, such attempts would allow using UML as an enterprise modeling language for developing computation neutral type of diagrams, which are more suitable for understanding and reasoning about enterprise system architectures [10]. It is recognized that

UML support for such task is vague, because semantic integration principles of different diagram types are lacking [20].

This paper demonstrates how service interactions can be integrated with behavioral and structural aspects of conceptual representations. The well-known example of Ford's case [19] is used to explain the benefits of the service-oriented approach [15], which was developed for conceptual modeling of information systems. We are not discussing advantages of service-oriented enterprise architectures [23], because they are quite obvious [26]. Nevertheless, given the central role of service concept in this study, we will focus on what is essential and most stable core in conceptual modeling of enterprise system, which is composed of the organizational or technical parts. Conceptual models of services should make sense not just for IS designers in the implementation-oriented domains, but also for business modeling experts, which are focused on computation neutral analysis of organizations. The presented service-oriented modeling approach shares many similarities with ontological foundation of service process [9]. The internal behavior of services is described by using basic principles of an ontological framework, which is developed by Bunge [3].

2 Value Flows and Service Interactions

Service flow modeling is quite intuitive way of system analysis that is easy to understand for business professionals and information system designers. Value models clarify why actors are willing to exchange business objects with each other. Actions are required for exchanging the physical, decision and data flows. Actions and exchange flows can be viewed as fundamental elements for defining business scenarios. A scenario is an excellent means for describing the order of service interactions. Scenarios help the enterprise system architects to express business processes in interplay with elementary service interactions between enterprise system components. Each service can be analyzed separately as is required by the principle of separation of concerns. In such way, value flows and service interactions provide a natural way of process decomposition.

Value flows are special types of concepts that represent moving things such as information or materials. Solid rectangles will be used for denotation of physical flows and light boxes will indicate data flows. Actions are performed by actors, which will be represented by ellipses. Actions are necessary for transferring flows between subsystems, which are represented as organizational actors. Actors will be represented by square rectangles. Two value flows between customer and vendor are illustrated in figure 1.

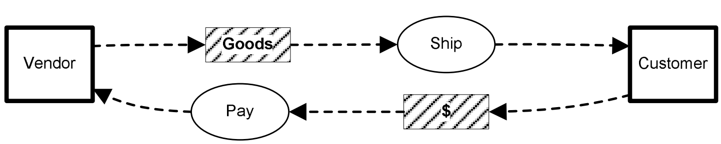


Fig. 1. Value flows between a Customer and Vendor

Direction of the material flow of Goods indicates that the Ship action is initiated by Vendor. Customer is a recipient of Goods flow. In contrary, money flow (represented by $) is initiated in exchange to Goods by Customer with the help of the Pay action. If so, then Vendor is the recipient of money flow. An action with a missing data or physical flow component will be understood as a decision or control flow.

Organizational actor is a unit of functionality exposed to environment that is conceptualized for representation of an enterprise subsystem. It depends on the individual actor role and his goals whether he considers some received information, materials or decisions as value flows. So, value flow is relative notion. Ideally, any elementary interaction must be motivated by the resulting value flow and vice versa a value flow typically requires initiation of some coordinating interactions, which are necessary for provision of value flows. Service interactions take place between service requester and service provider. In the DEMO theory [5], service requesters and service providers are viewed as active elements, which are represented by individuals or subjects. The DEMO approach distinguishes between two kinds of actions: production acts and coordination acts. Coordination acts are normally initiated by service requesters. Coordination acts are necessary to make commitment regarding the corresponding production act, which is supposed to bring a value flow to service requester. Production acts are normally performed by service providers. Therefore, they should be always associated with value flows.

Interaction dependencies can be used to conceptualize services between various enterprise system actors. Since all actors are implemented either as organizational or technical system components, they can use each other according to the prescribed service interaction patterns to achieve their goals. The simplest service interaction pattern can be represented by two interaction dependencies into opposite directions between service requester and service provider [16]. It is very similar to a well-known communication action loop. This idea is illustrated graphically in figure 2.

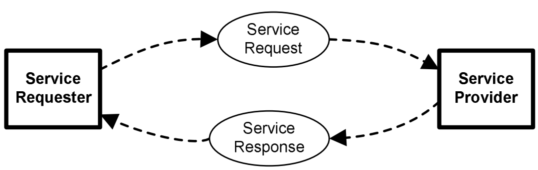


Fig. 2. Elementary service interaction loop

Interaction loop between two actors indicates that these two depend on each other by specific actions. Service providers are actors that typically receive service requests, over which they have no direct control, and initiate service responses that are sent to service requesters. Service requests and service responses can be used to define more complex interaction loops. In this way, designers are able to construct a blueprint of interacting components, which are represented by various types of actors across organizational and technical system boundaries. An enterprise system can be defined as a set of interacting loosely connected components [16], which are able to perform specific services on request. One example of service interaction loop, which consists of two interaction flows in two opposite directions, is presented in figure 3.

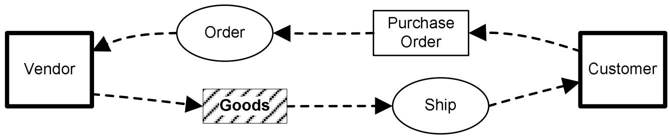


Fig. 3. Example of the elementary service interaction loop

This example can be considered as graphical description of elementary service interaction loop between Customer and Vendor. It defines two actions, which are available to the organizational actors. A customer may be interested to order goods. A vendor is able to ship goods. Coordinating actions can be viewed by actors as possibilities. Set of possibilities, which are delegated to one organizational actor, may help to understand his organizational power. By using such actions, the actors are entering into commitments regarding their obligations. For instance, the Order action can be viewed as a possibility for sending a purchase order by Customer to Vendor. If a Vendor would accept it, then he is obliged to ship Goods to the customer. In this way, service interactions can be used to analyze possibilities and obligations of the enterprise actors.

3 Enterprise System Decomposition Principles

Enterprise systems can be analyzed as compositions of technical and organizational components, which can be represented by various types of actors. Organizational components are interacting subsystems such as individuals, organizations and their divisions or roles, which denote groups of people. Technical components are represented by enterprise subsystems such as machines, software and hardware. We distinguish between two types of concepts: active and passive [17]. An actor can only be represented by an active concept. An instance of an actor is an autonomous subsystem. Its existence can only be motivated by a set of interaction dependencies with other actors that keep enterprise system viable. Passive concepts are represented by entities or classes of objects, which represent data at rest. All passive concepts are characterized by mandatory attributes. Objects of passive concepts can be affected by interacting subsystems, which are able perform specific services on request. Passive concepts are not the focus of this chapter, because they have no influence on system decomposition principles.

The ontological foundation of our modeling approach is based on the set of semantic dependencies between active concepts, which are viewed as enterprise actors. Actors are represented by subsystems that are non-overlapping in functionality. Active concepts can be related by the static dependencies such as inheritance, composition and classification [15]. Graphical notation of semantic dependencies, which are used to represent decomposition of enterprise system, is represented graphically in figure 4.

Fig. 4. Notation of static dependencies between actors

Classification dependency (●—) specifies objects or subsystems as instances of concepts. Classification is often referred to as instantiation, which is reverse of classification. It should be noted that classification of the object-oriented approaches is more restricted relation. It can be only defined between an object and a class. A class cannot play a role of meta-object, which is instantiated in another class, in the conventional IS modeling approaches. Instances of concepts propagate to the more generic concepts in the inheritance hierarchy. This feature is defined by the following rule:

If A ●— B and B ➡C then A ●— C.

Here: ➡ is inheritance dependency and ●— is classification dependency.

Actors have their own instances, which play role of subsystems. For example, Ford Motor Company is an instance of an Organization. Instances can be basic or derived. These two types of actor instances are represented in figure 5.

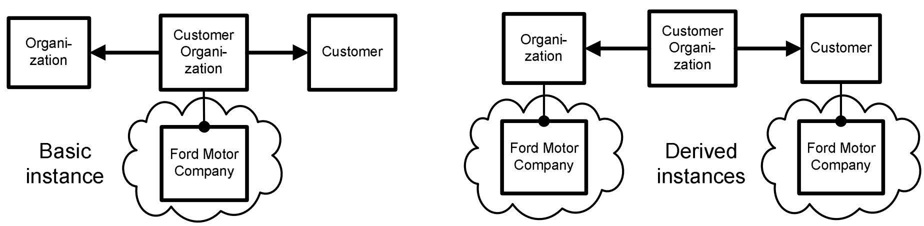


Fig. 5. Basic and derived instances of actors

For example, if (Ford Motor Company ●— Customer Organization) and (Customer Organization ➡ Customer) then (Ford Motor Company ●— Customer). Every instance of an actor is subsystem, which is characterized by the set of possibilities and obligations that are defined for this actor. Conceptual models of service interactions can be used to analyze possibilities and obligations, which are represented by the interaction dependencies. For instance, the Order action can be viewed as a possibility for sending a purchase order by Ford Motor Company to Vendor. If Vendor would accept it, then it is obliged to ship Goods to Ford Motor Company (see figure 3).

Composition (–►–), which is presented in this paper, differs significantly from the object-oriented composition [17]. A–►–B indicates that objects of the concept A are viewed as parts of composite B. is It is a much stricter form of aggregation, which allows just either 1 or 1..* multiplicities between wholes and parts. Other cases of conventional composition are not legal. It means that a part cannot be optional. Very important distinctive features of the presented composition dependency, that differentiate it from the other types whole-part dependencies, are as follows: (1) part is existent dependent on a whole, (2) if a whole has a single part, then this part has coincident lifetime with a whole, (3) if a whole has more than one part, then the creation of a first part is coincident with creation of a whole, creation and removal of additional parts can take place any time, but the removal of a last part is coincident with removal of a whole [14], (4) coincident creation and removal of a whole together with all parts is a special case of composition, (5) part can belong to only one whole of the same type. It should be noted that the optional cases of aggregation and composition can be always transformed into this stricter type of composition with the help of special modeling techniques [17].

Composition hierarchies can be used for detection of inconsistent interaction dependencies between actors. If the actors are loosely coupled, they never belong to the same decomposition hierarchy. Interaction dependencies of loosely coupled actors on the lower level of decomposition hierarchy are propagated to compositional wholes. So, composition links can be used for reasoning about derived interaction dependencies between actors on the higher granularity levels of conceptualization. Interaction dependencies between actors, which are placed in two different composition hierarchies, are characterized by the following inference rules:

1) If A –▶– B and R(A ····▶C) then R(B ····▶C),
2) If A –▶– B and R(C ····▶A) then R(C ····▶B).

Interaction dependency R(B ···▶C) between two actors B and C indicates that subsystem denoted by B is able to perform action R on one or more subsystems of C. Two subsystems of Customer Organization together with their interaction dependencies are represented in figure 6.

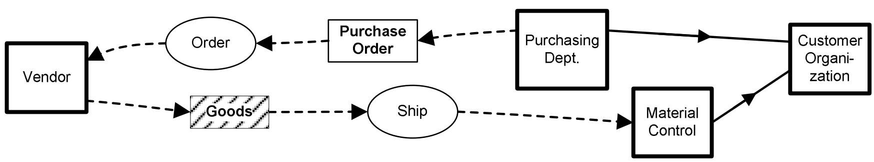


Fig. 6. Service interaction loop on the lower level of decomposition

Service interaction loops, which are specified on different granularity levels, must be consistent. Actors, actions and flows of consistent service interaction loops cannot be contradictory. For instance, if (Purchasing Department –▶– Customer Organization) and Order(Purchasing Department ····▶Vendor) then Order(Customer Organization ····▶Vendor). Note that service interaction flows, which are represented in figure 7, are consistent with the interactions of the previous diagram.

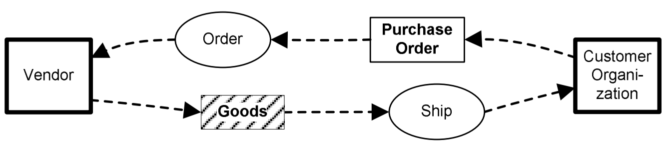


Fig. 7. Service interaction loop on the higher level of decomposition

Inheritance dependency between two actors indicates that these actors share structural and interactive similarities. More specific actors inherit the composition and interaction dependencies from more generic actors. It should be noted that in the object-oriented approaches, the inheritance link is defined just for associations, attributes and operations. Inheritance dependency is convenient for sharing service interaction loops of more generic actors. Interaction dependencies are inherited according to the following inference rules:

1) If A ➡ B and R(B ····▶C) then R(A ····▶C),
2) If A ➡ B and R(C ····▶B) then R(C ····▶A).

If (Customer Organization ➡ Customer), then Customer Organization inherits all service interaction links, which are specified for the actor of Customer (see figure 3). For example, if (Customer Organization ➡ Customer) and Order(Customer ····▶Vendor) then Order(Customer Organization ····▶Vendor). Customer Organization has possibility to send a purchase order to a Vendor and Vendor is obliged to ship goods to the Customer Organization. Please note that possibilities and obligations of these two actors must be consistent with the interaction flows, which were already presented in figure 6 and 7. It is not difficult to find out that the presented dependencies are consistent with the service request (Order) and response (Ship) of figure 3. They can be specified as follows: if Order(Customer Organization ····▶Vendor) then Ship(Vendor ····▶Customer). This example is based on a well-known case study [19]. Ford Motor Company (see figure 5) is playing a role of customer organization, which is decomposed into two departments (Purchasing and Material Control). Customer Organization concept is additionally defined as the specialization of two other active concepts such as Customer and Organization. Therefore, same service interaction loop, which is discussed in the previous chapter (see figure 3), is consistent with the other diagrams of the lower level of granularity (see figure 6 and 7). The only difference here is that initiation of the Order action is delegated to a Purchasing Department and acceptance of the Ship action is delegated to Material Control.

All derived service interaction loops of the enterprise system cannot be in conflict with the other dependencies, which are represented by a set of diagrams on various levels of abstraction. Inconsistency can be detected by identifying action or flow naming conflicts. For instance, two service interaction loops, which are presented in figure 8, are inconsistent.

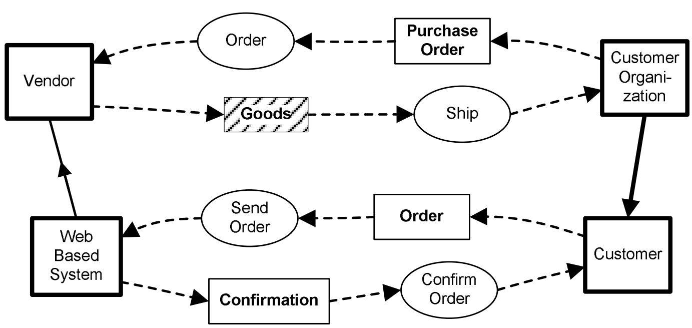


Fig. 8. Two inconsistent service interaction loops

Adding more specific actors into an enterprise system specification, must be justified by additional possibilities or obligations, which can be specified in terms of complementary interaction dependencies. It must be noted that the presented inference rules are useful, but not sufficient for reasoning about consistency of service interactions, which are specified on different levels of abstraction. To understand deep semantic structure of service interactions, the behavioral and related structural aspects of communication actions must be analyzed. How to express the interplay between business interactions and object transition effects is demonstrated in the next chapter.

4 Interplay of Structural and Behavioral Aspects of Service Architectures

Good principles of decomposition are not sufficient for systematic analysis of enterprise architectures. To understand the internal service interaction effects between service requesters and service providers, graphical representations of actions must be integrated with data descriptions. The dynamic aspects of service interactions can be expressed by defining structural changes of passive concepts. Passive concepts are classes of objects that are characterized by mandatory attributes. Values of such attributes are created or consumed in service requests and service responses. Structural aspects of data at rest are fundamental for understanding the semantic aspects of service choreographies and service architectures. From an ontological perspective, when two subsystems interact [3] one may affect the state each other. Changes in object state are characterized in terms of object properties. Interaction dependency R(A····►B) between two active concepts indicates that A subsystem can perform action R on one or more B subsystems. Actions are able to manipulate object properties. Property changes may trigger objects transitions from one class to another.

The internal effects of objects can be expressed by using transition links (——►) between passive concepts. There are three fundamental ways for representing object behavior by using reclassification, creation and termination actions [16]. If termination and creation action is performed at the same time, then it is called a reclassification action. Graphical notation of these three variations of actions is graphically represented in figure 9.

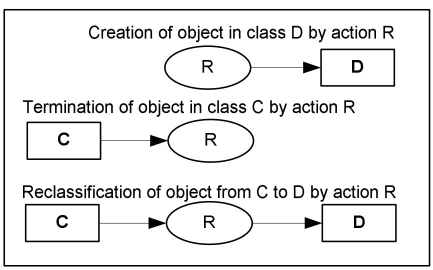


Fig. 9. Three variations of transition dependencies

Creation action is represented by a transition to an initial class. Termination action can be represented by a transition link from a final class. For instance, initiation of the Order action is typically used to create a Customer Order in a Vendor database or archive (see the diagram in figure 10). If Customer Order is accepted, then it may be used for triggering the shipping action. Creation and termination actions can be expressed by using object flows. A diagram showing operations and object flows with states has most of the advantages of activity diagrams without most of their disadvantages [1]. A transition arrow to action or transition arrow from action represents a control flow. In this way, any communication action can be used to superimpose interactions and control flow effects in the same diagram.

To demonstrate how service interactions can be used to analyze service architectures, we take well-known example of Ford's case [19]. The problematic situation of enterprise system was described as follows: *When Ford's purchasing department wrote a purchase order, it sent a copy to accounts payable. Later, when material control received the goods, it sent a copy of the receiving document to accounts payable. Meanwhile, the vendor sent an invoice to accounts payable. It was up to accounts payable, then, to match the purchase order against the receiving document and the invoice. If they matched, the department issued payment. The department spent most of its time on mismatches, instances where the purchase order, receiving document, and invoice disagreed. In these cases, an accounts payable clerk would investigate the discrepancy, hold up the payment.* The effects of described actions are expressed by using passive classes, which are represented in figure 10.

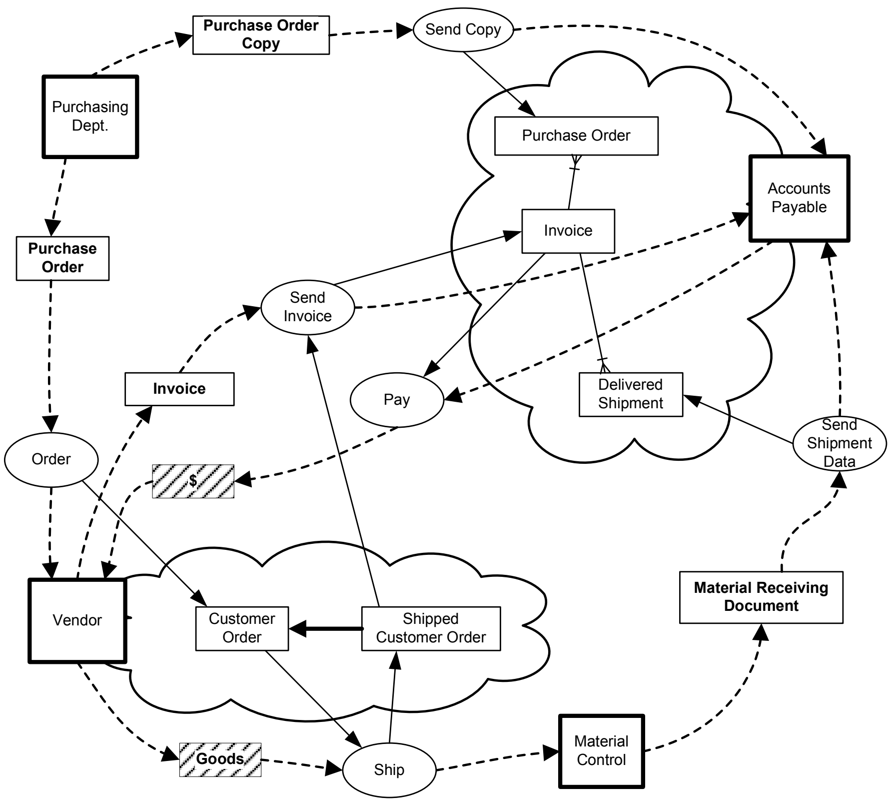


Fig. 10. Service architecture before radical change

This diagram represents conceptualization of two service loops. Purchasing Department is entitled to order goods by sending a purchase order. If it is accepted by vendor, then a Customer Order is archived. Vendor is obliged to ship goods for every outstanding Customer Order (service loop 1). Creation of Shipped Customer Order object is necessary precondition to generate another service request. It is used to trigger Send Invoice action. If Invoice flow is accepted by Accounts Payable, then the Pay action must be initiated (service loop 2). There are two explicit post-conditions for creating an Invoice. Every Invoice object must be linked with at least one Purchase Order and with at least one Delivered Shipment (see two mandatory associations from Invoice). Purchase Order object cannot be created without involvement of Purchasing Department. Creation of Delivered Shipment object is triggered by Material Control. This diagram also indicates that Send Copy and Send Shipment Data actions must precede Send Invoice action.

Most IS methodologies are not integrating interactive and behavioral aspects of service architectures, which were illustrated in the previous diagram. Failure to visualize interaction and object transition effects in a single diagram makes it difficult to understand the essential semantic differences of a deep business process structure. This example demonstrates how interactions should be linked together with structural changes of passive classes of objects. Transition dependencies between passive concepts help designers to reason about sequences, alternatives, synchronizations or iterations of service interaction flows [16]. We are not going into details of various control patterns for the reason of space limitations.

Structural changes of objects are manifested via object properties. Properties can be understood as mandatory attribute values. If diagrams are used to communicate unambiguously semantic details of a conceptualized system, then optional properties should be proscribed [12]. In the presented method, the attributes are defined by the total single-valued ($\rightarrow$) or by the total multi-valued attribute dependency ($\twoheadrightarrow$). If $A \rightarrow B$ or $A \twoheadrightarrow B$, then concept A is viewed as a class and concept B - as a mandatory property of A. One significant feature of our approach is that the association ends of static relations are nameless. Association ends cannot be used to denote mappings in two opposite directions. Semantics of dependencies are defined by cardinalities, which represent a minimum and maximum number of objects in one class (B) that can be associated to objects in another class (A). Expression $A \twoheadrightarrow B$ can be used to represent the following cardinalities: (0,1;1,1), (0,*;1,1) and (1,1;1,1). Expression $A \rightarrow B$ corresponds to (0,1;1,*) or (1,1;1,*) cardinalities. Graphical notation of an attribute dependency between A and B is represented in figure 11.

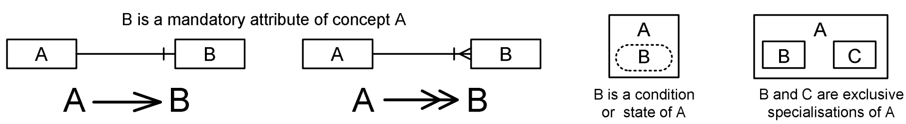


Fig. 11. Notation of attribute dependencies

Any passive concept can be defined as an exclusive complete generalization of two concepts. Passive concept can also be characterized by state [7] or condition [16]. Example of exclusive generalization and example of state is presented in the next diagram.

We will apply our approach to illustrate service architecture after the radical change in Ford Motor Company. The reengineering of the enterprise system in Ford was possible to make in two ways. These two alternatives are described as follows [19]:

One way to improve things might have been to help the accounts payable clerk investigate more efficiently, but a better choice was to prevent the mismatches in the first place. To this end, Ford instituted 'invoice-less processing'. Now when the purchasing department initiates an order, it enters the information into an online database. It doesn't send a copy of the purchase order to anyone. When the goods arrive at the receiving dock, the receiving clerk checks the database to see if they correspond to an outstanding purchase order. If so, the receiving clerk accepts them and enters the transaction into the computer system. If receiving can't find a database entry for the received goods, it simply returns the order. Computation neutral graphical description of the new situation is represented in figure 12.

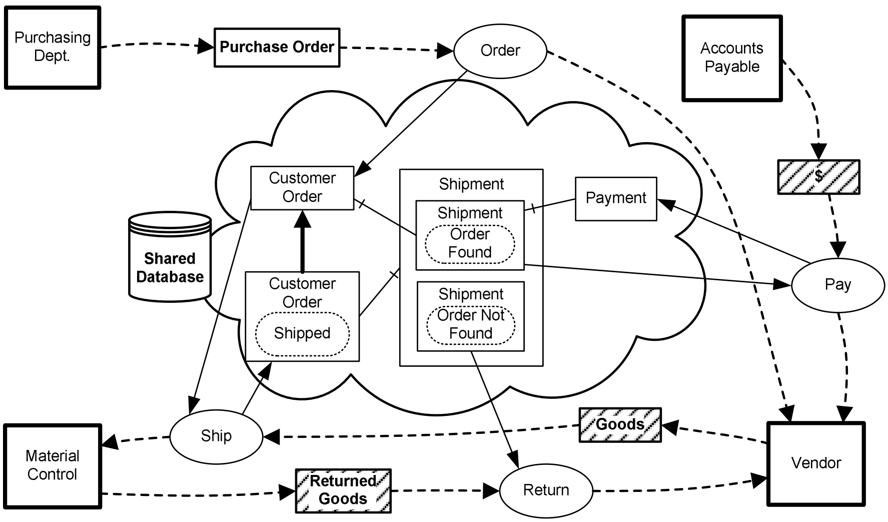


Fig. 12. Service architecture after radical change

Concepts and their static dependencies of a shared database are represented by cloud (it is used just for illustration purposes). Please note that this cloud is an implementation specific notion, which should not be represented in this diagram. Any interaction dependency represents creation (Order action), termination (Return action) or reclassification (Ship and Pay actions). Fundamentally, two kinds of changes occur during reclassification: removal of an object from a precondition class and creation of an object in a postcondition class. Object creation without any properties does not make sense. Sometimes, postcondition class may require preservation of some precondition class object. Such requirements can be specified by the static dependencies. Examples of such preserved classes of objects are Customer Order and Shipment [Order Found]. Preservation of Customer Order is required by the postcondition of Customer Order[Shipped] object (see Ship action). Creation of Payment requires preservation of the Shipment[Order Found] object. Quite often objects should not be preserved like in the presented case. They are passing several classes and then are destroyed.

A missing postcondition class represents termination, which is a special case of the reclassification. Termination action is supposed to remove all object properties, which are represented by attributes of a precondition class. In the presented diagram, the Return action not only initiates Returned Goods flow, but it also removes Shipment[Order Not Found] object. Note that Shipment is a property of Customer Order[Shipped]. So, if Shipment is removed, then the related Order[Shipped] object must be removed as well. The creation action is represented with a missing precondition class. Creation must bring all object properties, which are represented by attributes of a post-condition class, into existence. Creation event is illustrated by the Order action. Please note that this diagram is incomplete. Customer Order properties are not shown.

5 Discussion and Concluding Remarks

Service interaction dependencies are essential for understanding similarities of business processes. However, interaction flow analysis is not sufficient for studying the fundamental semantic differences of processes when reengineering the organizations. The structural and behavioral aspects of service architectures need to be analyzed in terms of object properties and object transitions between various types of classes. For instance, the essential similarities and differences of service architectures, which are presented in figure 10 and figure 12, can be identified by analyzing conceptualizations of two situations in Ford Motor Company. They illustrate not just interactions, but also some related behavioral and structural aspects of service architectures.

Essential similarities of service architectures are not difficult to identify by the help of inference rules, which are presented in this paper. The similarities of two situations in Ford are graphically expressed by the diagram, which is represented in figure 13.

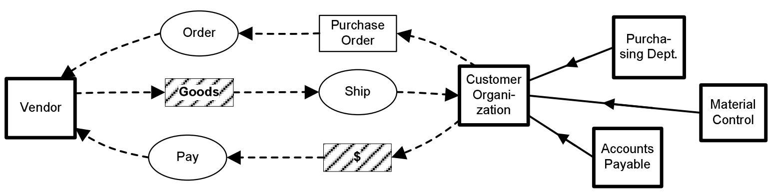


Fig.13. Essential similarities before and after radical change

This diagram characterizes in a concise form the basic enterprise system structure, which is defined in terms of actors and their interaction dependencies. The overlapping of interactions before and after the radical change, demonstrate that the generic business process structure is stable. Analysis of the same case with DEMO approach confirms stability of the essential model as well [6]. However, our service-oriented diagrams demonstrate some important differences between two situations. It is not difficult to understand that one service loop before the radical change is different. Pay action follows Ship action in both diagrams (see figure 12 and 10). However, the main difference is in Send Invoice action. In the service architecture before radical change, one complementary interaction is necessary, because an Invoice class object cannot be created directly by the Ship action (see figure 10). Invoice is required as a precondition for triggering of the Pay action. The related service requests and responses before the radical change can be expressed as follows:

1) if Order(Customer Organization ····▶ Vendor)
   then Ship(Vendor ····▶ Customer Organization),
2) if SendInvoice(Vendor ····▶ Customer Organization)
   then Pay(Customer Organization ····▶ Vendor).

If Send Invoice action before radical change could be replaced by Ship action, then triggering events and effects would be equivalent to what is represented in figure 13. The first service interaction loop is exactly the same in both situations and the second service loop is quite different. After the radical change, this service loop is as follows:

if Ship(Vendor ····▶ Customer Organization)
then Pay(Customer Organization ····▶ Vendor).

There is one more essential difference in the service architecture after radical change. It clearly reveals an alternative interaction flow of Returned Goods, which is missing in the architecture before radical change (see figure 12). New service requests and responses can be expressed as follows:

if Ship(Vendor ····▶ Customer Organization)
then Pay(Customer Organization ····▶ Vendor),
otherwise Return(Customer Organization ····▶ Vendor).

The alternative prescribes what happens if Pay action fails. Please note that Return action is missing in service architecture before radical change, because conditions on returning of goods were unclear (see textual description of the situation). After radical change, Shipment is defined as non-overlapping generalization of two concepts with mutually exclusive conditions. If newly created shipment order cannot be found (figure 12), then the Return action must be triggered by Material Control. Most information system design methodologies are 'blind' in expressing such alternative behavior, because they fail to combine together interactions with structural and behavioral aspects of service architectures. Use case, activity and state diagrams are difficult to integrate in a straightforward way. Structural analysis methods provide some possibilities for modeling components, data flows and processes on different levels of decomposition, but resulting hierarchies are clumsy.

Visual analysis of semantic differences between two conceptualizations reveals that the service architecture before radical change is more complicated in a variety of ways. Success of Ship action (with condition Order Found) is not sufficient for triggering the Pay action by Accounts Payable. Pay action before radical change requires creation of an Invoice flow by Vendor. It is obvious from the presented diagram (see figure 10), that Invoice action may fail more often, because of some other reasons:

1) Failure to trigger Send Copy action by Purchasing Department or to receive Purchase Order Copy by Accounts Payable (Purchase Order object is not created),
2) Failure to trigger Send Shipment Data action by Material Control or to receive the Material Receiving Document by Accounts Payable (Delivered Shipment is not registered).
3) Failure to link Invoice object with at least one Purchase Order or to link it with at least one Delivered Shipment in the Accounts Payable archive.

According to the presented diagram, these three situations cannot take place in the service architecture after radical change, because unnecessary coordinating interactions are removed and each Shipment object must only be associated with exactly one Customer Order (see figure 12). It is unclear how these semantic differences can be expressed by the DEMO method [5], because it is not taking into account interplay between service control flows and static dependencies of conceptualizations. To the best of our knowledge, expressing parallel, alternative and iterative behavior of service interactions in combination with structural properties is a challenge for most business process modeling methods.

Conceptual modeling methods, which put into foreground active concepts, are typically focusing on analyzing interactivity between organizational and technical components. The DEMO approach as well as the enterprise modeling language ArchiMate [23] are stemming from a similar modeling tradition. It is quite successful in analyzing an external behavior of enterprise system components. In contrary, the object-oriented paradigm is based on modeling of the static and dynamic aspects of passive concepts, which are represented by various classes of objects. Majority of the textbooks in the area of systems analysis and design recommend concentrating first on domain modeling [1], which is based on analyzing classes, attributes, relationships, state transitions and related effects. Most conventional system analysis and design approaches put into foreground modeling of passive concepts. Only few emerging approaches are making attempts to express deep semantics of interplay between active and passive structures [7]. Conventional information system analysis and design methods are projecting the structural, interaction and state-transition aspects into totally different types of diagrams. It makes very difficult verification of integrity between the static and dynamic aspects of various enterprise architecture dimensions. UML individual diagram types are clear enough, but integrated semantics among models is missing. That is why object-oriented diagrams are difficult to apply for business logics alignment with design for making both organizational and technical system parts more effective.

References

1. Blaha, M., Rumbaugh, J.: Object-Oriented Modelling and Design with UML. Pearson, London (2005)
2. Booch, G., Rumbaugh, J., Jacobsson, I.: The Unified Modelling Language User Guide. Addison Wesley Longman, Inc., Massachusetts (1999)
3. Bunge, M.A.: Treatise on Basic Philosophy. Ontology II: A World of Systems, vol. 4. Reidel Publishing Company, Dordrecht (1979)
4. De Marco, T.: Structured Analysis and System Specification. Prentice Hall, NJ (1979)
5. Dietz, J.L.G.: DEMO: Towards a Discipline of Organisation Engineering. European Journal of Operational Research (128), 351–363 (2001)
6. Dietz, J.L.G.: The Deep Structure of Business Processes. Communications of the ACM 9(5), 59–64 (2006)
7. Dori, D.: Object-Process Methodology: A Holistic System Paradigm. Springer, Berlin (2002)
8. Evermann, J., Wand, Y.: Ontology Based Object-Oriented Domain Modeling: Representing Behavior. Journal of Database Management 20(1), 48–77 (2009)
9. Ferrario, R., Guarino, N.: Towards an Ontological Foundation for Service Science. In: Domingue, J., Fensel, D., Traverso, P. (eds.) FIS 2008. LNCS, vol. 5468, pp. 152–169. Springer, Heidelberg (2008)

10. Finkelstein, C.: Enterprise Integration Using Enterprise Architecture. In: Linger, H., et al. (eds.) Constructing the Infrastructure for the Knowledge Economy, pp. 43–82. Kluwer Academic/Plenum Publishers (2004)
11. Gane, C., Sarson, T.: Structured System Analysis. Prentice Hall, NJ (1979)
12. Gemino, A.: To be or maybe to be: An empirical comparison of mandatory and optional properties in conceptual modeling. In: Proc. Ann. Conf. Admin. Sci. Assoc. of Canada, Information Systems Division, Saskatoon, pp. 33–44 (1998)
13. Glinz, M.: Problems and Deficiencies of UML as a Requirements Specification Language. In: Proc. of the 10-th International Workshop on Software Specification and Design, San Diego, pp. 11–22 (2000)
14. Guizzardi, G.: Modal Aspects of Object Types and Part-Whole Relations and the de re/de dicto distinction. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 5–20. Springer, Heidelberg (2007)
15. Gustas, R., Gustiene, P.: Pragmatic – Driven Approach for Service-Oriented Analysis and Design. In: Information Systems Engineering - from Data Analysis to Process Networks. IGI Global, USA (2008)
16. Gustas, R., Gustiene, P.: Service-Oriented Foundation and Analysis Patterns for Conceptual Modelling of Information Systems. In: Information System Development: Challenges in Practice, Theory and Education, vol. 1, Springer, Heidelberg (2009)
17. Gustas, R.: A Look behind Conceptual Modeling Constructs in Information System Analysis and Design. International Journal of Information System Modeling and Design 1(1), 79–108 (2010)
18. Gordijn, J., Akkermans, H., van Vliet, H.: Business Process Modelling is not Process Modelling. In: Mayr, H.C., Liddle, S.W., Thalheim, B. (eds.) ER Workshops 2000. LNCS, vol. 1921, pp. 40–51. Springer, Heidelberg (2000)
19. Hammer, M.: Reengineering work: Don't Automate, Obliterate. Harvard Business review, 104–112 (1990)
20. Harel, D., Rumpe, B.: Meaningful Modeling: What's the Semantics of 'Semantics'? IEEE Computer, 64–72 (October 2004)
21. Johannesson, P., Andersson, B., Weigand, H.: Resource Analysis and Classification for Purpose Driven Value Model Design. International Journal of Information System Modeling and Design 1(1), 56–78 (2010)
22. Krogstie, J., Lindland, O.I., Sindre, G.: Defining Quality Aspects of Conceptual Models. In: Proceedings of the IFIP8.1 Working Conference on Information Systems Concepts (ISCO3): Towards a Consolidation of Views, Marburg, Germany, March 28-30, pp. 216–231. Chapman & Hall, London (1995)
23. Lankhorst, M., et al.: Enterprise Architecture at Work. Springer, Berlin (2005)
24. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding Quality in Conceptual Modelling. IEEE Software 11(2), 42–49 (1994)
25. OMG: Unified Modeling Language Superstructure, version 2.2. (2009), `http://www.omg.org/spec/UML/2.2/` (retrieved January 19, 2010)
26. Steen, M.V.A., Strating, P., Lankhorst, M.M., ter Doest, H., Iacob, M.E.: Service-Oriented Enterprise Architecture. In: Service-Oriented Software System Engineering: Challenges and Practices, pp. 132–154. Idea Group Inc., USA (2005)
27. Yourdon, E., Constantine, L.L.: Structured Design. Prentice Hall, NJ (1979)
28. Zachman, J.A.: A Framework for Information System Architecture. IBM Systems Journal 26(3) (1987)

Exemplifying a Framework for Interrelating Enterprise Architecture Concerns

Sabine Buckl, Florian Matthes, Christopher Schulz, and Christian M. Schweda

Chair for Software Engineering of Business Information Systems (sebis),
Technische Universität München,
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl,matthes,christopher.schulz,schweda}@in.tum.de
http://wwwmatthes.in.tum.de

Abstract. In recent years, enterprise architecture (EA) management has gained increasing attention as means to support enterprises in adapting to changing markets and in seizing new business opportunities. A multitude of approaches and frameworks making prescriptions on how to document the different states of the EA have been developed. These approaches target different purposes and correspondingly different concerns (areas of interest) in the architecture. Hence, an enterprise seeking to develop or evolve an organization-specific EA documentation technique most likely runs into difficulties to understand the interdependencies between different frameworks and approaches.

The paper addresses the aforementioned challenge by presenting a framework for EA concern interrelations, which can be used to systematically analyze the concern relationships of different approaches. The applicability of the framework is demonstrated by means of a case study from the automotive industry, in which the framework is used for the development and enhancement of an EA description for risk management.

1 Introduction and Motivation

The increasing frequency of change, modern enterprises face in today's globalized and competitive environments, leads to a rising internal complexity of the socio-technical system *enterprise*. A promising and commonly accepted instrument to deal with this complexity and to foster business-IT-alignment is enterprise architecture (EA) management [1,2,3]. Originating from the field of information systems architecture (cf. [4]), EA management takes a holistic perspective targeting all areas of an enterprise from business and organizational via application and information to infrastructure and data aspects. EA is thereby in the sense of the ISO Standard 42010 understood as the "fundamental organization of a system [enterprise] embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [5]. A multitude of methods to EA management has been developed by researchers (cf. [6,7,2,8,3]), practitioners (cf. [9,10,11]), and standardization bodies (cf. [12]). Although differing in respect to the selected scope, reach, and focus, the

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 33–46, 2010.

proposed methods usually distinguish the following activities of the EA management function: a) document and maintain the current state of the EA, b) develop and describe planned and target states of the EA, c) enact and communicate planned EAs and architectural principles, and d) analyze and evaluate architectures.

A central challenge arising during the aforementioned typical activities of EA management is *stakeholder involvement* [12,13]. To address this challenge the architectural documentations referring to the current, planned, or future states of the EA, need to represent the corresponding concerns of the stakeholders. Thereby, concerns are defined in accordance with the ISO Standard 42010 as "those [areas of] interests which pertain to the system's [enterprise's] development, its operation or any other aspect that are critical or otherwise important to one or more stakeholders" [5]. Put in other words, a concern can be understood as the area of the enterprise that the respective stakeholder is interested in. Whereas, a product manager for instance is interested in the performance of the business process and the services responsible for creating the product, an application manager is concerned with the standard conformity of the business applications.

In the holistic perspective of EA management the two aforementioned concerns are clearly interrelated. Crosspoints are the terms service and business application. The thereby denoted concepts are most likely related – a service is provided by business applications – or may even be identical – two terms referring to the same concept. The differences in terminology can be explained by the different language communities the two stakeholders belong to. To develop a comprehensive EA description, covering the different concerns of the stakeholders, the relations between concerns should be made explicit. Furthermore, these relations between concerns could be utilized to decide on a maturity roadmap for the evolution of the EA management function. In an initial step an enterprise might decide to stick to a more simple concern, while after first results have been achieved the concern is expanded to support a more detailed investigation. This directly yields the research questions that our paper addresses:

- *How does a framework for interrelating EA concerns look alike?*
- *What types of relationships between EA concerns exist?*
- *How can the framework be utilized in developing and evolving EA models?*

In the remainder of the article above research questions are answered by developing a conceptual framework for EA concern relationships. The framework is derived and presented in a step-wise manner (see Section 2). A real world case study form the automotive industry is utilized to illustrate the applicability of the framework in answering the aforementioned research questions in Section 3. Complementingly, prominent approaches from literature to EA management in general and EA modeling in special are revisited to ensure comprehensiveness of the approach (see Section 4). In Section 5, we conclude with a critical reflection of the achieved results and an outlook on future research directions.

2 A Framework for EA Concern Relationships

In line with the ISO 42010 [5], descriptions of architectures are comprised of viewpoints that conversely reflect areas of interest in the corresponding architecture. These areas of interest, called *architectural concerns* in the terms of the ISO 42010, not only define the part of the architecture that should be considered, but also bring along a conceptualization of the EA. Put in other words, a concern builds and centrally employs an underlying domain ontology targeting a specific part of the overall EA. Considering two or more different architectural concerns, it may be beneficial to understand, in which ways the corresponding areas of interest relate. In order to facilitate in-depth discussions on concern relationships, we subsequently present are more precise definition of what an architectural concern means. A concern can be described by two fundamental properties, namely

- a **selection** of relevant architectural elements, i.e. of *things* and their *properties* in an ontological sense, that are in the area of interest, and
- a **conceptualization** describing the architectural *types* and *property types* abstracted from the selected (relevant) architectural elements.

Illustrating above definition of an architectural concern in the EA context, we formulate a Concern A that describes the

> *business applications in relationship to their responsible organizational units* (conceptualization) of an enterprise, *restricted to organizational units located in Germany* (selection).

In the light of the above definition, we can derive different types of relationships between concerns. A basic relationship expresses the fact that two concerns have at least one type in common, i.e. that their conceptualizations share at least one type. This entails a more sophisticated discussion on the question of how to understand 'share' in this context. In more detail, two ways to understand the term exist: *syntactic* and *semantic*. While sharing in the latter sense means that two concerns have a type with similar meaning in common, syntactic sharing describes that two concerns are related via a concept with a common name. To qualify these two different understandings of the term 'share', we use the term *intersection* to denote syntactic sharing.

Based on the intersection-relationship, we establish the notion *compatibility* as a relationship between two concerns, indicating that all intersecting types and property types of two concerns are also semantically shared. If in contrast at least one intersecting type or property type is assigned distinct meanings in both concerns, the concerns are called *incompatible*. Introducing a Concern B describing

> *business applications and their standardization level* (conceptualization) of an enterprise,

we can exemplify the aforementioned concern relationships in the following. Both concerns are clearly intersecting, as they both employ the type 'business application'. At this point nothing can be said on the compatibility of these concerns.

To answer this question, one would have to delve into the contexts, in which these concerns were raised. With concerns being described as sentences in real language, such investigation would employ analyses of the linguistic communities (cf. Kamlah and Lorenzen [14]) from which the concern descriptions originate. We abstain here from going into the details of interpersonal testing techniques, but call for an intuitive understanding of *equivalent meanings*, based on the assumption that each type described in the concerns was backed by a definitory sentence in a glossary or an ontologically richer structure. Based on the glossary, equivalence of meaning can be understood as stakeholder consensus in accordance to Kamlah and Lorenzen [14]. Two types thereby share a common meaning, if "any informed member of [both] linguistic communities would [...] say so". Given the fact that both concerns referred to a single definitory glossary, explaining what a business application is, we can state that the two concerns are compatible.

Grounded on the basic relationship of compatibility, we establish a *super*-relationship between architectural concerns. One concern is superconcern of another, if the concerns are compatible and all types and property types of the second are intersecting ones. Put more colloquially, the conceptualization of the first concern completely covers the conceptualization of the second one. Introducing another Concern C that describes the

> *business applications in relationship to their responsible organizational units and their supported business processes* (conceptualization) of an enterprise.

We further assume that Concern A and Concern C are *maximally* compatible, i.e. that all shared concepts each have an equivalent meaning. Under these premises, Concern C completely covers Concern A, exemplifying superconcern relationship with Concern C being super to Concern A.

Above, we introduced three types of concern relationships based on an intuitive understanding of the concern's underlying conceptualization. In the context of EA management such conceptualizations are frequently described as object-oriented meta-models, see e.g. Buckl et al. in [15], Johnson and Ekstedt in [16], and Österle et al. in [17]. Subsequently, we shall call these meta-models in accordance with Buckl et al. *information models*. Based on such more formally described conceptualizations the aforementioned types of concern relationships can be defined more concisely and described in more detail. We nevertheless abstain from discussing the subtleties of such definitions, an application of the concern relationships along exemplary information models is given in Section 3. Some additional remarks on concern relationships based on their corresponding conceptualizations are described by Buckl et al. in [18].

Concluding the exposition of the concern relationship framework in this section, we discuss on relationship types that ground in the selection that constitutes a concern. The first relationship type to devise, is the one of *instance level intersection*. Two concerns are related in this way, if they – applied to the same EA – cover at least one common element, i.e. thing, from the architecture. Refraining the examples from above, we display CONCERN D that describes the

organizational units (conceptualization) of an enterprise.

Assuming that both CONCERN A and CONCERN D have a common understanding what the term "organizational unit" means, the concerns are intersecting on instance level. Complementing, we can define a *non-intersection* relationship. In addition, we can derive an *instance level superconcern* relationship, expressing that one concern – applied to the same EA – covers all elements that another concern – applied to this EA – covers.

We complement the above framework with a discussion on the utilization of the relationships in the context of EA management. Refraining an argument put forward by Aier et al. in [19], an enterprise-specific EA description technique must match the requirements of the using enterprise in two directions, namely in "width" and "depth". Width in this context means, that the technique must cover the relevant parts of the EA embracingly, i.e. must be able to answer questions of all relevant EA stakeholders. Complementing, the criterion of depth accounts for the fact that for answering certain EA questions more detailed information about the EA may be needed. While in the context of initially devising an EA description technique, multiple different approaches, also ones not using concern relationships, might be helpful, we expect that the relationships especially simplify the evolution of the EA description technique in response to changing demands of the EA stakeholders. To illustrate how this can be achieved, we build on the EA management pattern catalog of Technische Universität München [20,21]. This catalog expatiates a set of practice-proven EA relevant concerns that are complemented with information models reifying the underlying information demands. If the concerns were related using the aforementioned relationship types, an enterprise willing to evolve its own EA description technique based on the information models would have the following advantages:

- Feeling the need to "widen" the scope of the EA description technique, the enterprise could search the set of concerns that are *compatible* with the currently employed ones. These concerns and their complementing information models should easily integrate into the current description technique.
- Feeling the need to "deepen" the scope of the EA description technique, the enterprise could traverse the *superconcern*-relationships, in order to determine concerns that employ the currently described concepts, but cover additional information.

Summarizing the above, concern relationships provide decision support for developing an enterprise-specific EA description technique. Illustrating this fact, we subsequently provide an application example from a practice case, where we applied concern relationships to prepare decision making on the appropriate EA description technique for addressing an EA challenge in both 'right width and depth'.

3 Application Example

An international manufacturer from the automotive industry intends to construct a new overseas production plant. Thereby, the local market should be served in a customer friendlier manner, while currency fluctuations have to be mitigated in order to remain competitive. Part of this large-scale project is the deployment and roll-out of approximately 170 partly interconnected applications supporting a modern and up to date assembling of vehicles. For the sake of profitability, existing applications are duplicated and slightly adjusted to the local conditions before being deployed and tested. These applications are typically referred to as *group applications*, meaning they are used in a slightly adapted manner at different production plants. Thereby, those new applications are also connected with their counterparts on the other side of the globe, enabling a company-wide business intelligence, monitoring, and controlling capability.

The local production and sales of vehicles is constrained by country-specific laws and regulations. In the case of our international car manufacturer, the disclosure of company-internal information is regulated by the *antitrust law*. Once a piece of information is revealed to one supplier or customer, it must be automatically made accessible to other third parties. Contrastingly, if a *class action suits* succeeds, hence an objection raised by several plaintiffs is sustained, lawyers will have the possibility to access different company-internal information as well.

As a matter of fact, information is mostly stored within applications nowadays allowing an effective and efficient data organization and retrieval. These applications directly and indirectly support the execution of different business processes (e.g. car construction process, car sales process, etc.). Due to above alluded rules and regulations, external lawyers and auditors would have unimpeded access to those applications as well as their contained data. As second consequence, the attorneys were also allowed to retrieve information not only stored locally on those applications, but also on other applications which are connected via network technology. Hence, the real set of information being also transitively accessible through the 170 applications would be much broader than anticipated before, representing a danger for corporate secrets.

To respond to this challenge, the international manufacturer launched a project targeting at modifying those applications also granting access to data, which is not explicitly needed for the production plant. Initially, 35 applications as well as their exported and imported interfaces providing access on data not relating to the overseas factory where identified and marked as inherently critical. The classification thereby was mostly based on the experience application owners and users had with those applications. Subsequently, the 35 applications classified as critical will undergo time-consuming analysis and complex functional modifications aiming at restricting their data scope to the information relevant for the local context. The remaining 135 applications not classified yet, will be analyzed in a two step approach. To reduce workload, a quick but comprehensive interface analysis of the 135 applications will be performed in a first step to identify those applications, which need to be considered in more detail. In a second step, the applications classified as critical will be analyzed as described before.

When being confronted with the application interface analysis, the car manufacturer asked us for help. The problem description of our cooperation reads as follows:

> How can criticality of application systems interfaces be determined? Which concepts and relationships are decisive for criticality of interfaces exported by applications.

After thoroughly studying a list of factors all determining if an interface is critical, we suggested three different concerns, which are detailed in the following.

As a first step, we generally defined an interface as being critical whenever it has no authorization mechanism available. Hence, once an interface possesses such a mechanism, its criticality is set to low and it needs no further investigation. Figure 1 illustrates the underlying information model I-1 consisting only of one single entity named `interface` and the attributes `hasAuthorization`, `hasAuthentication`, and `criticality`. Thereby, the attribute `criticality` is a derived attribute, whose value is influenced by the two other attributes. This fact is illustrated utilizing the notation of *feature dependency relationship* as defined by Buckl et al. in [22].

An alternative way to assign a specific criticality to an application interface is to closely examine the security type of the data the interface exchanges. This can be expressed through a second concern, which is described in the information model I-2 depicted in Figure 2 by showing two entities, `interface` and `data`. Thereby, an `interface` is regarded as being low critical whenever it is used for exchanging public data only. If the data is either internal, confidential, or secret, the interface's criticality shifts to medium or high. Again the notation of the *feature dependency relationship* (cf. [22]) is used.

Compared to our aforementioned framework for interrelating EA concerns, both models are compatible. The entity `interface` is semantically and syntactically equal within the two concerns, both models could complement each other.

After further consideration, we proposed a third concern whose information model I-3 also addresses the goal of determining critical and non-critical interfaces. Thereby, an interface may only be deemed as being critical when it is from the type 'exporting' with regards to the interface's underlying application.

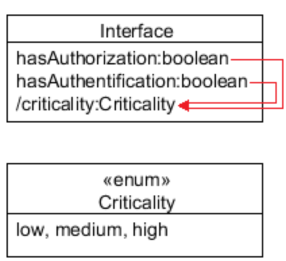


Fig. 1. Interface criticality defined by an available authentication mechanism

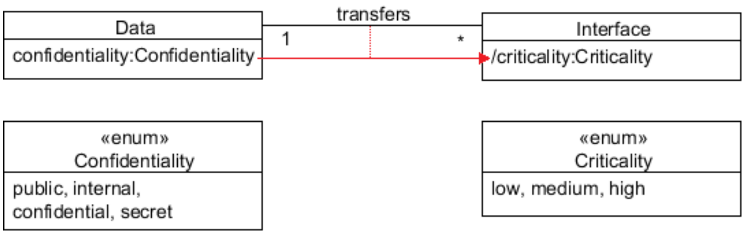


Fig. 2. Interface criticality contingent on the security type of exchanged data

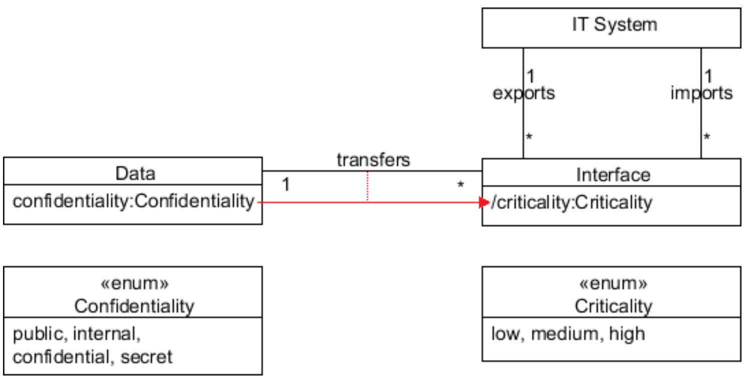


Fig. 3. Interface criticality depending on interface and exchanged data security type

In addition, the interface has to transfer non-public data, thus data which is either internal, confidential, or secret.

As expatiated on in Section 2, Figure 3 represents a super concern with regards to the concern specified in Figure 2. Additional relationships between information model I-1 and I-2, respectively I-3 exist. Each of the above pair describes two compatible concerns. The interrelations between the different concerns worked out so far, are depicted in the map presented in Figure 4. In the map, we further added an "artificial" concern I-3', as the integration of I-1 and I-3.

When discussing the results with the international car manufacturer, two further fields of work arose. Firstly, we identified additional attributes, which may be regarded as being relevant for the criticality of an application and its

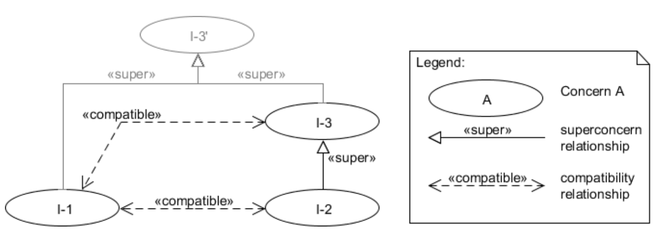


Fig. 4. Concern map for the exemplary concerns

interfaces, i.a. multi-client capability on different application layers, concrete data content (i.e. business objects and their criticality), configurability of the interface, etc. Those aspects could be either directly incorporated in existing information models or new concerns had to be created. Secondly, we had to find out which of the required information addressing the concerns of application interface criticality presented above is already stored in existing EA management tools. If the specific date is not available it has to be gathered, e.g. through interviews with the according application owners or an online survey. In all, the goal of determining the criticality of application interfaces was addressed through several interrelating EA concerns, summarized by a concern relationship map, which follows the presented framework.

4 Concern Relationship in the State-of-the-Art in EA Descriptions

In addition to the first successful application of the framework for EA concern relationship in practice as discussed in the preceding section, we analyzed existing literature on EA management. EA management is a research topic with an increasing number of publications [23], in which especially the fields of EA modeling and EA analysis are heavily researched. Central to both activities is the notion of the architectural description reflecting the corresponding architectural concerns. Subsequently, we explore the state-of-the-art in EA modeling and analysis with a special emphasis on the EA descriptions used thereby as well as their underlying concerns in order to validate the framework for interrelating EA concerns as presented in Section 2.

The approach of *multi-perspective enterprise modeling (MEMO)* was initially presented by Frank in 2002 [6]. Therein, Frank outlines a modeling framework, which is based on an extendable set of special purpose modeling languages. The special purpose modeling languages correspond to the different language communities, which typically exist in an enterprise, e.g. salespersons or project managers. By further providing a common meta-language, the *MEMO meta modelling language (MML)* [24], which the special purpose languages rely on, the integration of the different languages can be facilitated. Examples for special purpose languages are e.g. the *strategy modelling language (MEMO-SML)* [25] and the *organization modelling language (MEMO-OrgML)* [26]. Concepts of two different languages can be associated, i.e. are common to both languages. Therefore, the different concerns addressed by the MEMO special purpose languages can be regarded as being *intersected* and *compatible*.

The communication challenge already mentioned in the approach presented by Frank, is further discussed by Buckl et al. in [27], Schelp and Winter in [28], and Schönherr in [13]. While the first publication refers to the communication challenge within an enterprise, the latter two publications discuss the different language communities mirroring the academic groups conducting research in the area of EA management. Due to the absence of a standardized terminology and a commonly-accepted description language for EA management, the different approaches and communities have developed their own terms, leading to *incompatibility* issues, when different approaches are combined.

A systemic perspective on EA modeling is presented by Wegmann et al. in [29]. They provide a method and a tool to formalize the alignment of the multiple levels that constitute an EA. In particular, they propose to organize the different concepts that constitute the EA in *organizational* and *functional levels*. Thereby, the functional levels represent behavioral and the organizational levels the constructional hierarchy. Within each organizational level, two different viewpoints are available, the *information viewpoint* – a black box view on the respective concern – and the *computational viewpoint* – a white box specification of the concern. These viewpoints can be refined, which results in a hierarchy of viewpoints and the underlying information, respectively. Based on this understanding, a *super-* and *sub-relationship* between different viewpoints and concerns can be identified. Furthermore, Wegman et al. point out that a "vocabulary mapping" [29] between related concepts on different organizational levels has to be performed, which can be ascribed to the already mentioned *compatibility* question.

In [7] Kurpjuweit and Winter present a *systemic approach to meta model engineering*, which consists of five steps: 1) Identification of relevant concerns, 2) requirements elicitation, 3) viewpoint relationship overview, 4) meta model[1] fragment selection or design, and 5) meta model fragment integration. For the identification of relationships between concerns, especially the last step is of importance. Kurpjuweit and Winter allude to two different types of relationships between concerns as mirrored in the corresponding meta model fragments. Firstly, concerns can be *intersected*, which rises the challenge that the terminology must be adjusted. Secondly, a generalization or specialization may be necessary, which is caused by a *superconcern* or *subconcern* relationship between different concerns.

In [16] Johnson and Ekstedt discuss an approach to EA decision making based on EA models and analyses. Based on the current documentation of the EA, future EA scenarios are derived and assessed in respect to selected quality attributes, as e.g. performance, interoperability, availability, security, or usability. For each of these quality attributes, an *influence diagram* is presented by the authors, which details on causal dependencies between architectural properties. If for instance availability is considered, the reliability or recoverability of the system under consideration influences the overall availability. Thus, the concerns included in an influence diagram have a *subconcern/superconcern* relationship to each other, while concerns of different influence diagrams may be *intersected*.

Central ideas for a language for describing EAs are outlined by Jonkers et al. in [30] and further elaborated in [31]. They discuss two requirements that relate to the notion of the concern, namely *meta model flexibility* and *integration of heterogeneous models*. Meta model flexibility is thereby meant to demand that general EA description concepts can be refined to organization-specific concepts and standards. Conversely, a language for describing EAs must facilitate the

[1] The term *meta model* thereby corresponds to the term *information model* as used in this paper.

integration of heterogeneous models, as special purpose languages for specific parts of the EA exist, whose concepts can be 'translated' into or 'associated' with concepts for EA description. Jonkers et al. further discuss in [30] the distinct levels of detail, on which the corresponding descriptions may act. In this sense, one or more concerns from special purpose modeling languages might be 'aggregated' into a single concern at the EA description level. In [31] Jonkers et al. provide a core meta-model for architectural descriptions, more precisely behavioral aspects thereof, based on a dichotomy of *service-* and *implementation-aspects*. The corresponding basic concepts *service* and *behavior element* are refined to specific *business*, *application* and *infrastructure-layer* concepts, respectively. Nevertheless, the basic modeling language allows to omit concepts on some of these layers, if they are not of interest for the specific modeling purpose. Put in the terminology of our concern-discussion, the language implicitly allows for the derivation of *subconcerns*.

Summarizing the state-of-the-art in EA concern relationship modeling it can be stated, that the framework presented in Section 2 can be used to classify the different relationships as explicitly or only implicitly proposed in current publications. Furthermore, it provides a framework for conceptual interrelation as well as proposes the *concern map* for making the relations explicit.

5 Conclusion and Outlook

In this paper, we discussed the concept *concern* in the context of EA modeling and EA management, respectively. Concerns thereby represent distinct information demands in respect to the described architecture and can hence be reflected by information models that provide a 'schema' for the information to be modeled and collected during the corresponding EA management process. Based on this model-centric understanding of concern, we developed a framework for interrelating EA concerns and showed its applicability in a real world case study from the automotive industry. Thereby, especially the utility for evolving modeling techniques was discussed. Complementingly, we revisited the state-of-the-art in EA modeling and analysis.

This paper presents itself as a research-in-progress paper, that outlines the idea of concern relationships. From this point different directions for further development and research are open. At first, the informal presentation of the relationship types in this paper should be complemented with a more formal understanding thereof. In this respect, the semantics of the different relationship types could be formally defined based on a formal understanding of architectural descriptions. This would then allow for more intricate considerations on the relationships between information models that reflect the corresponding concerns. Secondly, concern relationships might not only be an issue of theoretical interest, but may prove to be interesting for supporting maturity considerations for EA management functions. In this respect, the relationships could also be beneficial for evolving an already existing EA management function towards a 'broader' and 'deeper' coverage of the overall EA. This subject has yet not been investigated in practice, although the notion of the "related" concerns as outlined by Ernst in [32] and in [21] have shown to be beneficial during the process of establishing and evolving an EA management function.

Acknowledgement

The case study research presented in this paper was possible thanks to the support of Volkswagen AG, Wolfsburg. The authors of the document wish to acknowledge the industry partners for their help and important contributions.

References

1. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations. IBM Systems Journal 32(1), 472–484 (1993)
2. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Heidelberg (2005)
3. Ross, J.W., Weill, P., Robertson David, C.: Enterprise Architecture as Strategy. Harvard Business School Press, Boston (2006)
4. Zachman, J.: Extending and formalising the framework for information systems architecture. IBM Systems Journal 31(3) (1992)
5. International Organization for Standardization: Iso/iec 42010:2007 systems and software engineering – recommended practice for architectural description of software-intensive systems (2007)
6. Frank, U.: Multi-perspective enterprise modeling (memo) – conceptual framework and modeling languages. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), Washington, DC, USA, pp. 1258–1267 (2002)
7. Kurpjuweit, S., Winter, R.: Viewpoint-based meta model engineering. In: Reichert, M., Strecker, S., Turowski, K. (eds.) Enterprise Modelling and Information Systems Architectures – Concepts and Applications, Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007), Bonn, Germany, Gesellschaft für Informatik, October 8-9, LNI, pp. 143–161 (2007)
8. van der Raadt, B., van Vliet, H.: Designing the enterprise architecture function. In: Becker, S., Plasil, F., Reussner, R. (eds.) QoSA 2008. LNCS, vol. 5281, pp. 103–118. Springer, Heidelberg (2008)
9. Dern, G.: Management von IT-Architekturen (Edition CIO). Vieweg, Wiesbaden (2006)
10. Keller, W.: IT-Unternehmensarchitektur. dpunkt.verlag, Heidelberg (2007)
11. Niemann, K.D.: From Enterprise Architecture to IT Governance – Elements of Effective IT Management. Vieweg+Teubner, Wiesbaden (2006)
12. The Open Group: TOGAF "Enterprise Edition" Version 9 (2009), `http://www.togaf.org` (cited 2010-02-25)
13. Schönherr, M.: Towards a common terminology in the discipline of enterprise architecture. In: Aier, S., Johnson, P., Schelp, J. (eds.) Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research, Sydney, Australia, pp. 107–123 (2008)

14. Kamlah, W., Lorenzen, P.: Logische Propädeutik: Vorschule des vernünftigen Redens, 3rd edn. Metzler, Stuttgart (1996)
15. Buckl, S., Ernst, A.M., Lankes, J., Schneider, K., Schweda, C.M.: A pattern based approach for constructing enterprise architecture management information models. In: Wirtschaftsinformatik 2007, pp. 145–162. Universitätsverlag Karlsruhe, Karlsruhe (2007)
16. Johnson, P., Ekstedt, M.: Enterprise Architecture – Models and Analyses for Information Systems Decision Making. Studentlitteratur, Pozkal (2007)
17. Österle, H., Winter, R., Hoening, F., Kurpjuweit, S., Osl, P.: Der St. Galler Ansatz des Business Engineering: Das Core Business Metamodel. Wisu – Das Wirtschaftsstudium 2(36), 191–194 (2007)
18. Buckl, S., Matthes, F., Schweda, C.M.: Interrelating concerns in ea documentation – towards a conceptual framework of relationships. In: 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM 2010), Paderborn, Germany (2010)
19. Aier, S., Kurpjuweit, S., Saat, J., Winter, R.: Enterprise Architecture Design as an Engineering Discipline. AIS Transactions on Enterprise Systems 1, 36–43 (2009)
20. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2008)
21. Chair for Informatics 19 (sebis), Technische Universität München: Eam pattern catalog wiki (2010), `http://eampc-wiki.systemcartography.info` (cited 2010-02-25)
22. Buckl, S., Ernst, A.M., Schweda, C.M.: An extension to the essential meta-object facility (emof) for specifying and indicating dependencies between properties. Technical report, Technische Universität München (2008)
23. Langenberg, K., Wegmann, A.: Enterprise architecture: What aspect is current research targeting? Technical report, Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (2004)
24. Frank, U.: The memo meta modelling language (mml) and language architecture (icb-research report). Technical report, Institut für Informatik und Wirtschaftsinformatik, Duisburg-Essen, Germany (2009)
25. Frank, U.: Memo: Visual languages for enterprise modelling. Technical Report 18, Arbeitsberichte des Instituts für Wirtschaftsinformatik Koblenz (1999)
26. Frank, U., Jung, J.: The memo organisation modelling language (memo-orgml). Technical Report 26, Arbeitsberichte des Institut für Wirtschaftsinformatik der Universität Koblenz-Landau (2001)
27. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M.: State of the art in enterprise architecture management 2009. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2009)
28. Schelp, J., Winter, R.: Language communities in enterprise architecture research. In: DESRIST 2009: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, pp. 1–10. ACM, New York (2009)
29. Wegmann, A., Balabko, P., Lê, R.G., Rychkova, I.: A method and tool for business-it alignment in enterprise architecture. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 113–118. Springer, Heidelberg (2005)

30. Jonkers, H., van Burren, R., Arbab, F., de Boer, F., Bonsangue, M., Bosma, H., ter Doest, H., Groenewegen, L., Scholten, J., Hoppenbrouwers, S., Iacob, M.E., Janssen, W., Lankhorst, M., van Leeuwen, D., Proper, E., Stam, A., van der Torre, L., van Zanten, G.: Towards a language for coherent enterprise architecture descriptions. In: 7th International Enterprise Distributed Object Computing Conference (EDOC 2003), Brisbane, Australia, IEEE Computer Society, Los Alamitos (2003)
31. Jonkers, H., Goenewegen, L., Bonsangue, M., van Buuren, R.: A language for enterprise modelling. In: Lankhorst, M. (ed.) Enterprise Architecture at Work, Springer, Heidelberg (2005)
32. Ernst, A.M.: A Pattern-Based Approach to Enterprise Architecture Management. PhD thesis, Technische Universität München, München, Germany (2010)

Ontology Representations for Information Exchange between Communities

Benjamin Diemert, Marie-Hélène Abel, and Claude Moulin

Heudiasyc UMR 6599 - Université de Technologie de Compiègne
Centre de Recherches de Royallieu
BP 20529 60205 Compiègne cedex, France
{benjamin.diemert,marie-helene.abel,claude.moulin}@hds.utc.fr
http://www2.hds.utc.fr/

Abstract. Exchange and repurposing of digital information and content raise language variations issues between communities. Indeed, producer and consumer may not share the same practices and languages. TV producers tend to have more and more collaborative workflows and uses amateur contributions to diversify their content's sources. However, amateurs may not understand the audiovisual jargon. In this paper, we introduce a conceptual model and its OWL formalization to deal with these issues. We show that our approach extends the thesaurus approach and is able to attach various labels to any kind of ontological element (class, property, instance). We find our motivations in information exchange between professionals and amateurs as well as in multilingual search. We study digital archiving and cultural heritage model and use them as the foundations of our work. We show how these models tend to focus on material exchange and are not sufficient to deal with a comprehensible exchange of information. We propose to use personal and contextual parameters to make a selection of comprehensible label and other kind of information for a given user. We finish by demonstrating the OWL formalization of our model.

Keywords: knowledge representation, thesaurus, denomination, digital archiving, TV production.

1 Introduction

The massive introduction of digital content on public and private networks has considerably increased the amount of documents and information available. However, digitisation and network availability have not yet kept their promises to ease content's exploitation and repurposing [1]. The TV production field is not an exception. The task is even more difficult as the production process is segmented into various stages. From script writing through funding, shooting, editing to post-production, each stage has its own specialized know-how and jobs. Nevertheless, each stage re-use audiovisual content or information from the previous ones.

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 47–62, 2010.

Dedicated Information Systems have been developed to meet these various needs one step at a time. They usually focus on content processing and leave aside the transportation of associated information valuable for production, archiving and repurposing. Indeed, information such as compositional metadata, – describing the content's structure – descriptive metadata – describing the context of production – and the description of intentions may lead to significant efficiency gains to describe the content's and ease its production and repurposing [2].

The few metadata available are produced in various manners. Some are produced automatically by systems – GPS coordinates, camera lens information etc. Others are automatically extracted from the audiovisual content – shot detection, speech recognition etc. Some others are manually filled in by the various protagonists engaged in the production chain. These information are usually lost or trapped into proprietary formats. The lack of integration between systems leads to a loss of information as the production goes on [3]. Thus, we believe there is a strong need to design a system which can wrap up both content and metadata and transport them without loss throughout the production chain. The package will need to hold the required metadata for repurposing.

In addition to interoperability issues, the diversity of protagonists engaged in the TV production process leads to language variations (vocabulary, spelling, abbreviation etc.) in the production of information, which may be critical for content's description. Even though they may share common concepts there are always language differences which make the content's description even more difficult to use and understand for the one which is searching. This problem increases with the use of archive and content from other organizations. The issue could be defined at a terminology level where we need to consider various denominations for the same concept, each one having a given scope of use.

It is also essential to build up a model capable of integrating and join these language variations whatever the syntax or jargon used. This feature is needed in order to carry on content and its description along the chain and ensure that these information will be comprehensible and re-usable. An ontology could tackle this issue provided that it considers and connects communities with their own variation. Multilingualism is thus dealt as a specific case of a more generic concern.

In this article, we present the advantages of modelling denomination's variations for information exchange between community of users. We consider two use-cases, the first one with two communities having different level of understanding of the same jargon, the second one exemplifies a search in a multilingual corpus – see Section 2. We introduce then some related works at the roots of our model and explain why they are not sufficient to address the language variations issue – see Section 3. We define the key concepts and relations of our model and illustrate its principles through the modelling of our two use-cases – see Section 4. Eventually, we discuss our OWL representation with these two examples – see Section 5 – and conclude on future works.

2 Motivations

This work takes place in the MediaMap project[1] which make use of Semantic Web technologies to promote collaborative production and audiovisual content repurposing. The consortium includes the two Belgian public broadcasting network[2] as main users. Thus, multilingual aspects are more like a requirement than a feature.

Moreover, the arrival of new competitors from the telecommunication field urges these organizations to develop collaborative production, to diversify their content's supply source (shooting, internal archive, purchase) and find new opportunities to exploit their productions. Description of the organization's content, search and integration of content from other organizations and even amateurs become crucial needs. Exchange of content and related information has gone beyond the scope of the organization and take place at various stages of the production chain. As the chain becomes more and more versatile there is a strong need for indexation since pre-production and all along the chain in order to support these exchanges.

2.1 Amateur and Professional Collaboration

Calls for amateur's contribution constitute one promising way to purchase original and sometimes exclusive content. This raises however a quality issue as the content may not meet the professional quality requirements. One pro-active approach to address this issue is to provide the amateur with editorial guidance written by professionals. But a shooting script written by a director is not sufficient as the amateur may not have a sound knowledge of professional's jargon and know-hows. For instance, shot values such as *American Shot*, *Medium Close-Up* or *Medium Shot* refer to a given framing of a character which needs to be explained to the amateur cameraman – see Figure 1.

We face here an information exchange issue between two communities with different levels of knowledge of the same jargon. We indicate in this figure how one concept could be presented differently for a (**P**)rofessional cameraman and an (**A**)mateur one.

2.2 Multilingual Information Retrieval

In some cases, we may consider the multilingual issue [4] as spelling variations denoting the same concept or instance. A proper name may have various acknowledged spellings which all refer to the same individual. For instance, there are many spellings for the name of the *Swan Lake's* composer – see Figure 2. Suppose now a French user makes a search on the composer with the French spelling. English and Russian documents will not be part of the results set even though the user understands these languages. Unless all spellings refering to the same instance are connected together and the user's capabilities are described.

[1] http://www.mediamapproject.org/

[2] The Flemish Radio- and Television Network (VRT), the French-speaking Radio- and Television Network(RTBF).

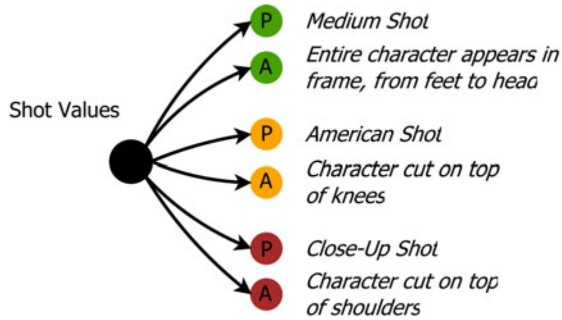


Fig. 1. A sample of shot values used in the audiovisual (**P**)rofessional jargon and their explanations for the (**A**)mateur

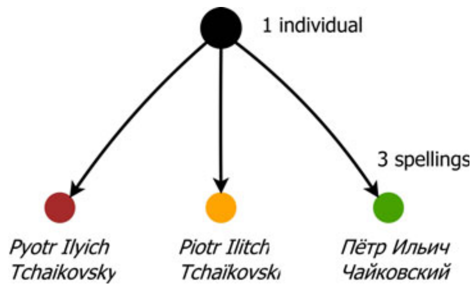


Fig. 2. English, French and Russian spellings for the proper name of the Swan Lake's composer

In order to connect these spellings we first have to distinguish them. Obviously, each spelling may be characterised by its natural language transcription. Also, the first two are written in the Latin alphabet, the last one is in Cyrillic. In other cases, we may eventually need to consider geographical locations to distinguish between local variations – English versus American spelling for instance.

3 Related Works

Among the works we have studied, *Acquisition, eXchange, Indexing and Structuring* (Axis) has brought us the best foundations to describe audiovisual description all along the chain. Axis is both a model and an architecture developed during the Memories project[3] to deal with exchange and archiving of media assets. Axis bases itself on principles brought by *Open Archive Information System* (OAIS), a digital archiving norm. We introduce in the next section the archiving principles of OAIS that have been used in Axis in order to address the exchange issue of media assets.

[3] http://www.memories-project.eu/

3.1 Digital Archiving and Information Exchange

Archiving may be defined in a documentary perspective as the intentional will to constitute and preserve a memory [6]. This is achieved through two indivisibles kind of preservation: (1) the material preservation of documents against physical deterioration, (2) the preservation of a reading tradition to convey the ability to interpret and understand the content of documents. Thus, archiving ensures us a sensible and comprehensible access to documents. We want to introduce in the next section the OAIS definition of archiving and its relation with information exchange.

Archiving as a service. The OAIS norm defines digital archiving as an information exchange service provided to various communities [5]. OAIS gives recommendations for setting up an organization which can provide such a service. Figure 3 shows the functions assumed by the organization as well as the data flows.

The main focus of this organization is to act as a middleman between a Producer and a Consumer community and take care of every aspects of the information exchange. In this perspective, *Preservation Planning* is charged to monitor changes in information representation technologies and practices in both communities. Additional information to ensure future exploitation and understanding is included in various *Information Package* (IP) which are used as conveyors. We detail first the information flow and then information requirements which define the IP's structure.

Information exchange management. The *Ingest* function is the first step of the chain. Its purpose is to gather the Producer's data and transform them in conformity with Preservation Planning quality requirements. A *Submission* IP (SIP) is defined conforming to previous agreements with the Producer. After ingestion, an internal format is used, the *Archival* IP (AIP). The information to be preserved is stored by the *Archival Storage*. Its description is handled separately

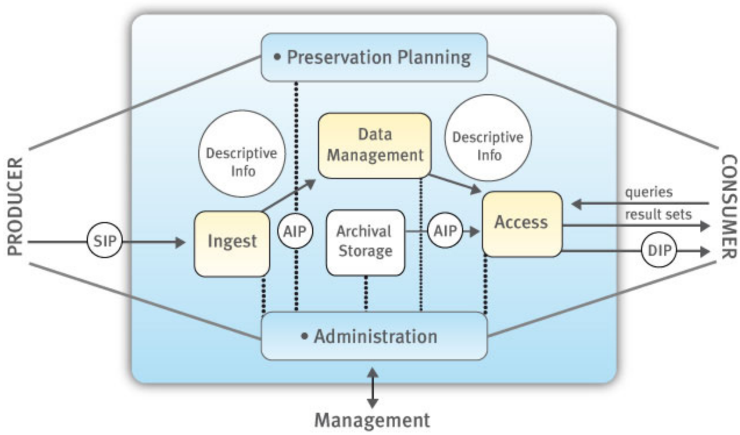


Fig. 3. OAIS functional organization and information flow

by the *Data Management* function to meet search needs. Indeed, the *Access* function uses these descriptions to handle the interaction – search, order, retrieve – with the Consumer community. Once the information needed is ordered, the AIP is transformed into a *Dissemination* IP (DIP). Notice that each IP is intended for a different use which may require format transformations or restructuring according to the general information requirements defined by OAIS.

Information's description requirements. OAIS defines four kind of requirements for its IP in order to ensure material and comprehensible access to information to be preserved:

- *Content Information* is the information to be preserved along with the descriptions needed for the Consumer community to exploit it. The descriptions deal with data structures, their meanings and the knowledge required to understand it. It could be as much detailed as needed, this decision belongs to Preservation Planing. For instance, a PDF file of a musical score may need first a plain text file for describing the PDF format, provided that the designated community is able to read such a file. Then another plain text file is required to explain the musical notation used.
- *Preservation Description Information* are identification and traceability information. Identification is needed for the whole and each part of the Content Information. Traceability starts with a description of the production's context, the author's original intentions, a processing history and integrity indicators.
- *Package Description* is used for search and exploration of the Content Information.
- *Packaging Information* is the information needed to bind and identify each part of the Content Information to its physical location on an information carrier.

As we have shown, OAIS define organizational and description requirements to handle information exchange between various communities. These requirements also consider archiving goals but seems to leave aside the issue of language variations between its communities. The issue is somehow raised but left to be solved by the organization and no modelling solution is expressed. Eventually, OAIS does not consider the specific case of media assets which raises other description issues. In the next section, we present a general model for cultural object representation.

3.2 FRBRoo

Functional Requirements for Bibliographic Records (FRBRoo) is a conceptual model dedicated to museum and library information exchange [7]. Its main purpose is to represent the relation between people engaged in the various stage of cultural object production, from the very first idea to the concrete realisation. Each cultural object is described on three levels:

– the abstract level of ideas that have no concrete existence outside an individual's mind, a melody or a story for instance (*Work* class in FRBRoo).
– the expression level where there may exist various possible expressions of the same idea (*Expression* class in FRBRoo). For instance a written novel, its translations and a screenplay adaptation of the novel.
– the carrier level which holds the various material manifestations of expressions (*Item, Manifestation Product Type, Manifestation Singleton* classes in FRBRoo). For instance, a book, a score or a cd-rom.

These basic distinctions are particularly helpful to describe the contribution of each individual in the production process of a media asset. It gives a very detailed view on the production context, so precise that it is hardly used in other context than cultural heritage. The Axis model represents media assets in pursuance of these distinctions and specialized them for its own use. The next section presents how Axis takes a stand on these norms and inspires us in our own work.

3.3 Axis

Axis is an implementation of OAIS which focuses on media assets description and exchange between organizations. As a consequence, Axis needs to consider the media industry concerns and specificities. Hence, the architecture has been exploded to consider more flexible information flows than proposed in OAIS. For instance, each organizations taking part in the flow may provide one or more functions, sometimes redundantly. Moreover, Axis deal only with one information package which holds for each kind of exchange (submission, archival, dissemination). The idea is that one package conforming to archival recommendations may also conform to interoperability requirements.

Concerning media assets description, the Axis model borrows the FRBRoo basic distinctions and use them to build its core ontology. Dedicated specializations are added in order to handle the specificities of temporal media such as audio or video. The base is to distinguish between a conceptual level (*Entity* in the Axis Core), a sign level (*Document* in Axis Core) and a physical or digital carrier level (*PhysicalClip* in Axis). Each element in a higher level may be related to several elements of the next lower level. For instance, one concept may be expressed in a variety of documents, each corresponding to a given facet of the concept. In a similar way, each document may be represented on several digital or physical carriers. For instance, the *Little Night Music* from Mozart has several interpretations played by various orchestra stored on various carriers.

On the sign level however, Axis restricts itself to a thesaurus approach to handle language variations. This approach enables for each concept to bind various alternate labels to a unique reference or preferred label in the same way that the W3C's *Simple Knowledge Organization System* (SKOS). The OWL Full representation of SKOS enables to define relations between concepts (broadMatch, narrowMatch, relatedMatch, exactMatch, closeMatch) to structure a scheme and enable mapping between them in a Semantic Web fashion [8].

The thesaurus approach does not seem sufficient for our concern, as it does not provide any way to characterize a label's scope of use, aside from the *xml:lang* attribute. Thus, natural language is the only hold to select a label for a given user. The automatic selection of a label pertaining to a specific jargon is not possible. Even though the label could be bound with the preferred label, there will still lack an attribute to characterize it.

In a way, we could say that the thesaurus objective is to normalize the use of language whereas our goal is to provide each community with a dedicated access to information – without losing the ability to exchange with and understand information from other communities. With SKOS thesaurus, the only manner to provide that kind of service is to define one thesaurus for each community and make the mappings between them. We define in the next section our modelling solution to overcome this issue.

4 Modelling

We first present the general principles which underly our modelling – section 4.1 – then we will illustrate them with the two use-cases previously described – section 2 for their descriptions, section 4.2 for their modelling. We finish by defining in details the main concepts and relations of our conceptualization – section 4.3.

4.1 Principles

We found with Axis a good model to deal with the majority of the MediaMap project issues. Hence, we have used their Core as the basis of our work. However, the thesaurus approach does not cover our concerns about language variations between community of users engaged in a production process. Our modelling extends the thesaurus approach to all kind of information and makes it more flexible so as to select the most comprehensible information for a given user engaged in a given context.

Our approach relies on the articulation of an information model, a user model and a context model. Figure 4 presents each part of the model. The idea is to bind a description of the information's production context to a description of the information access context, including a user description. Contextual parameters help us to select more precisely the most comprehensible information for a given user.

The selection of a denomination (instance of the Term class) is first made upon condition of the user's (instance of the PhysicalPerson class) personal abilities (instance of the Skill class). These abilities may refer to either a know-how or the knowledge of a jargon, a natural language, a syntax (instance of the Notation class). Other contextual parameters may help to make a finer selection. For instance, the membership in an organization (instance of the CorporateBody class) which may have its own practices – vocabulary, specific syntax etc. The job or position (instance of the Role class) is another indicator which may indicate the user's practical or linguistic knowledge. Locations may also help to identify the linguistic skills of a user. This example is extendable to other kind of information such as documents.

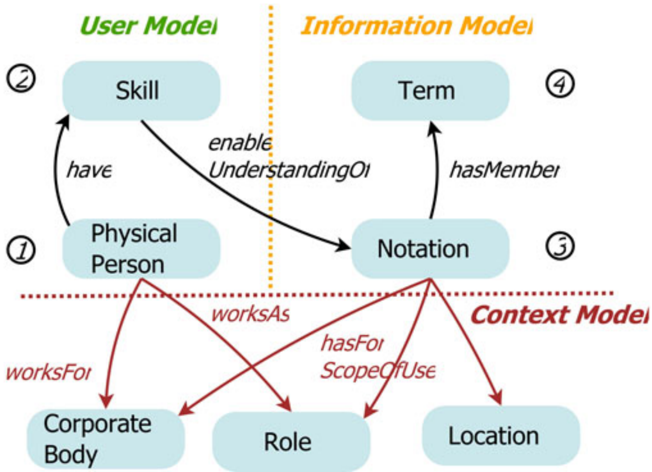


Fig. 4. Main concepts and relations between the User model, the Information model and the Context model

4.2 Use-Case Modelling

In this section we show the modelling of the use case of section 2, their OWL representation are described in section 5. The first use-case illustrates two French-speaking user, a professional cameraman and a amateur one, both accessing to the same script and in particular a shot value – see Figure 5. There are two distincts denominations for this shot value, the first is concise, the second one is more explanatory. The distinction between those two users is made on their skill level (amateur, professional) which gives them access to distinct Notations and thus a distinct or custom-tailored view on the script. Each denomination (instance of the Term class) is member of one of these Notations. Both denominations are related by a similarity relation when they refer to the same shot value. The Figure shows the connection between a given user and a given denomination.

The Tchaikovksi use-case illustrates the modelling of a multilingual search. Suppose one user requesting in French information about the famous composer. In this case, the request first brings us to the individual, to whom other pieces of information are attached. The real improvement comes from the user description – we know he understands both French and English – which enables us to bring a new source of information to the fore – English B document – aside from the French A document.

We may complicate the example by adding other attributes to the documents, such a character encoding parameters – ISO 8859-1 for Latin alphabet, ISO 8859-5 for Cyrillic alphabet. This kind of attributes may be useful when not only the user but also its device's abilities have been described. We define in the next section the concepts and relations of our model.

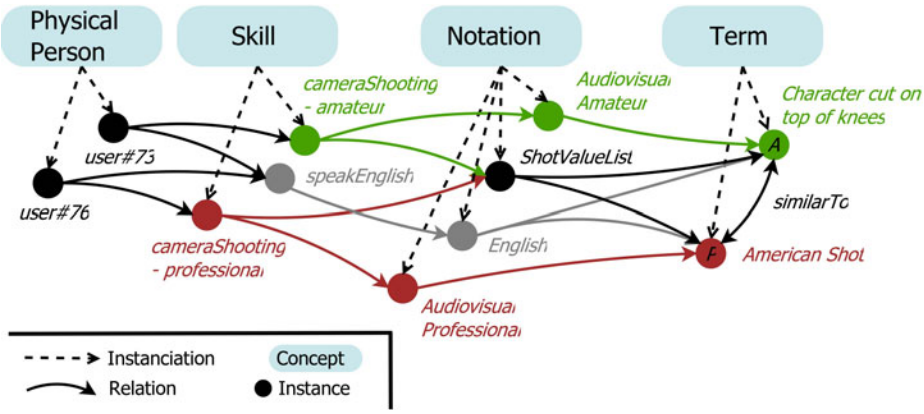


Fig. 5. Relations' network between users and denominations

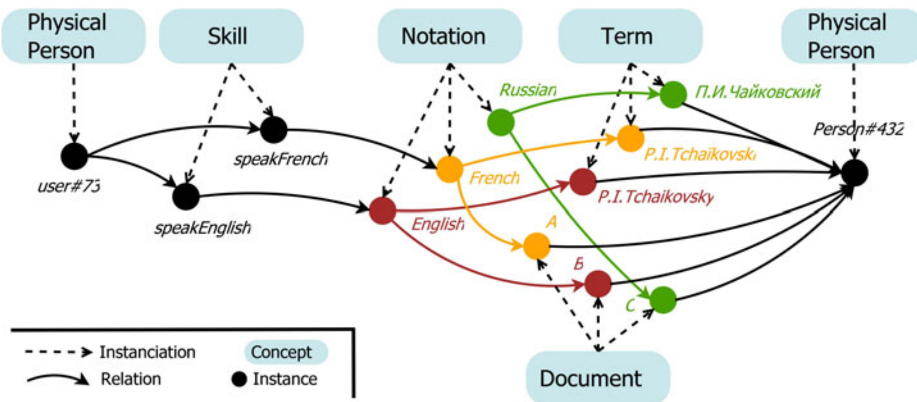


Fig. 6. Multilingual search and documents' selection

4.3 Conceptualization Definition

We divide the presentation of our conceptualization by its three different parts.

Information Model's concepts and relations. A term (instance of the Term class) wraps a property's value which would usually be considered as a string. We also attached various attributes to provide a description of its context of use. The real datatype of the value could be anything, word, date, number etc.

A document (instance of the Document class) denotes information related to a concept. Concepts are abstract entities which only exist through the material manifestations of documents. Contrary to FRBRoo, manifestation is not a concept but a relation (isAManifestationOf) between a document and an entity.

A notation (instance of the Notation class) designates a set of terms sharing common characteristics. They may refer to each other to build richer Notation and thus models the different aspects of a term. The Notation class have several specializations that we detail below:

- a naturalLanguage (instance of the NaturalLanguage class) models the membership of a term to a natural language – e.g. French, English, Flemish etc. Country code may be added to model geographical variations of a language – English & UK or English & USA.
- a syntaxEncodingScheme (instance of the SyntaxEncodingScheme class) models a specific way to encode bits. This include character encoding system (UTF-8), datatype encoding (xsd:string) or more complex syntax such as date format (ISO 8601 : 1988 (E)[4]) or even file formats (jpg, pdf).
- a vocabularyEncodingScheme (instance of the VocabularyEncodingScheme class) models a chosen set of terms such as a professional thesaurus. It may be used in a general way to model naming conventions within a business, organisational or geographic context. A metadata description scheme such as Dublin Core[5] is one example.
- an authorityList (instance of the AuthorityList class) models a finite list of terms which values have been arbitrarily chosen. For instance, the ISO 639-2 norm is a list of three-letters code referring to a language name, e.g. *eng* stands for english.

A term is member of a notation (isMemberOf relation) when it shares the corresponding characteristics. The membership is not exclusive so we may refine the description of a term at will. This relation is specialized by the natural language's membership (inLanguage relation). For instance, the value *Tchaikovsky* may be described both as member of the English naturalLanguage and encoded in ISO 8859-1. A notation is in conformity with (conformsTo) a SyntaxEncodingScheme when it uses its encoding rules. A notation may of course conform to several schemes.

User Model's concepts and relations. An agent (instance of the Agent class) models any acting entity. This class has three specializations:

- a physicalPerson (instance of the PhysicalPerson class) models an individual, e.g. *Tchaikovksy*.
- A corporateBody (instance of the CorporateBody class) models an organization composed of other agents, e.g. the *Minneapolis Symphony Orchestra*.
- A proxy (instance of the Proxy class) models a system acting on behalf of another kind of agent, e.g. a web server.

A skill (instance of the Skill class) models an agent's ability to perform a given task or to understand an information related to that task, e.g. setup instructions. This class has two specializations:

---

[4] W3C standard for date format, see http://www.w3.org/TR/NOTE-datetime for further details

[5] http://dublincore.org/documents/dcmi-terms/

– a linguisticSkill (instance of the LinguisticSkill class) is an agent's ability to understands a given natural language.
– a professionalSkill (instance of th ProfessionalSkill class) models an agent's ability to perform specific task and understand professional notations. For instance, *Framing* is an agent's ability to handle a camera given shot value instructions. These shot values are grouped together in an authorityList.

A physicalPerson is defined as skilled (isProficient relation) when the management considers him so.

Context Model's concepts and relations. Skills helps to define jobs or position (instance of the Role class). An agent – individual, organization or system – playing a given role is assigned a set of rights and duties in a project's scope. An individual may be employed by an organization (workFor relation) in a given position (workAs relation). A location (instance of the Location class) denotes a geographical or political entity. Location is a generic concept that may be specialized if needed.

Relations between the model's part. There are three major relations connecting each other part of the model. The first one connects a user with notations, the second one binds notations with contextual parameters and the third one links a term with any ontological element:

– a skill designates the ability of an agent to understand a notation (enablesUnderstandingOf relation)
– a notation may be used in various context (hasForScopeOfUse relation), an organization, a location or a role.
– a term may be attached to any ontological element to label it (isNamedBy relation), instance, concept or property.

5 OWL Formalization

The use of ontology seems to be profitable on several points. First, it requires the model's semantic to be explicit and well documented which will thus ease its understanding by the users and its maintenance by the designers. Moreover, the fact to focus on domain objects rather than a particular task may improve the flexibility and extensibility of the model [9]. When it comes to search on instances, query extension has proven valuable [10]. Eventually, reasoning capability and data integration between various models are particularly precious when dealing with information interchange and interoperability issues [11]. Knowledge Representation languages such as RDF and OWL offer us a way to express our ontology. We have chosen the OWL-DL species to ensure a good expressiveness without losing computational completeness. We discuss details of our formalization in the next section.

5.1 Denominations for Amateur and Professional

We present here an OWL representation of the shot value use-case in N3[6]. We have an authorityList with two terms referring to the same shot value. This shot value has no existence by itself, there is only these two terms which belongs either to an amateur or professional vocabularyEncodingScheme. On one side we distinguish the two terms by their membership in a different scheme (isMemberOf, inLanguage relations) and then on the other side we need to identify their equivalence (similarTo relation).

```
// ==== NOTATIONS ===== //
mm:shotValueList
      a       mm:AuthorityList ;
  mm:hasMember mm:AmericanShot-a, mm:AmericanShot .
mm:Audiovisual-Professional
      a       mm:VocabularyEncodingScheme ;
      mm:hasMember mm:AmericanShot .
mm:Audiovisual-Amateur
      a       mm:VocabularyEncodingScheme ;
      mm:hasMember mm:AmericanShot-a .
mm:English
      a       mm:NaturalLanguage .

// ==== TERM ===== //
mm:AmericanShot
      a       mm:Term ;
      mm:inLanguage mm:English ;
      mm:isMemberOf mm:shotValueList , mm:Audiovisual-Professional ;
      mm:similarTo mm:americainShot-a ;
      mm:value "American Shot"^^xsd:string .
mm:AmericanShot-a
      a       mm:Term ;
      mm:inLanguage mm:English ;
      mm:isMemberOf mm:shotValueList  , mm:Audiovisual-Amateur ;
      mm:similarTo mm:AmericanShot ;
      mm:value "A shot in which the character appears
       from head to knee"^^xsd:string .
```

The user's description is made through their skills and skill level. The skillLevel is an attribute which can take two values, amateur or professional. Therefore, each skill may have two versions, each of which enables to understand a different notation – here Audiovisual-Amateur and Audiovisual-Professional vocabularyEncodingScheme. However, each version may grant access to a common notation – here the shotValue authorityList.

```
// ==== PHYSICAL PERSON ===== //
mm:user_73
      a       mm:PhysicalPerson ;
      mm:isProficient mm:cameraShooting-a , mm:speakEnglish .
```

[6] http://www.w3.org/DesignIssues/Notation3.html

```
mm:user_76
      a       mm:PhysicalPerson ;
      mm:isProficient mm:cameraShooting-p , mm:speakEnglish .

// ==== SKILLS ===== //
mm:speakFrench
      a       mm:LinguisticSkill ;
      mm:enablesUnderstandingOf
              mm:English .

mm:cameraShooting-p
      a       mm:BusinessSkill ;
       mm:enablesUnderstandingOf
              mm:Audiovisual-Professional , mm:shotValueList ;
      mm:SkillLevel "professional"^^xsd:string .
mm:cameraShooting-a
      a       mm:BusinessSkill ;
      mm:enablesUnderstandingOf
              mm:Audiovisual-Amateur , mm:shotValueList;
      mm:SkillLevel "amateur"^^xsd:string .
```

5.2 Denominations and Multilingual Search

The multilingual use-case gives us the opportunity to talk about the OWL representation of the naming relation (isNamedBy). It has to be an Annotation Property so as to be attached to any kind of ontological element. Therefore, it is a good substitute for the *rdfs:label* relation. The OWL-DL species implies some constraints on the use of annotation property, but this has no impact in our case. Indeed, we only build triplet in the form `subject mm:isNamedBy term` where the subject may be an owl:class, an owl:property or an instance. The next example shows two terms carrying various spellings for Tchaikovski's proper name, and three documents related to the Composer (isAManifestationOf relation). There are also the natural language used.

```
// ==== NOTATIONS ===== //
mm:French-NL
      a       mm:NaturalLanguage .
mm:English-NL
      a       mm:NaturalLanguage .
mm:Russian-NL
      a       mm:NaturalLanguage .
// ==== TERM ===== //
mm:Tchaikovsky-EN-Term
      a       mm:Term ;
      mm:inLanguage mm:English-NL ;
      mm:value "Pyotr Ilyich Tchaikovsky"^^xsd:string .
mm:Tchaikovsky-FR-Term
      a       mm:Term ;
      mm:inLanguage mm:French-NL ;
      mm:value "Piotr Ilitch Tchaikovski"^^xsd:string .
```

```
// ==== DOCUMENT ===== //
mm:Document_A
      a       mm:Document ;
      mm:inLanguage mm:French-NL ;
      mm:isAManifestationOf
              mm:Tchaikovsky .
mm:Document_B
      a       mm:Document ;
      mm:inLanguage mm:English-NL ;
      mm:isAManifestationOf
              mm:Tchaikovsky .
mm:Document_C
      a       mm:Document ;
      mm:inLanguage mm:Russian-NL ;
      mm:isAManifestationOf
              mm:Tchaikovsky .
```

An individual's name may have several variations (isNamedBy relation). We have here one user having two different linguisticSkill which may eventually help us to select two comprehensible documents (A and B) out of three – see section 4.2.

```
// ==== PHYSICAL PERSON ===== //
mm:Tchaikovsky
      a       mm:PhysicalPerson ;
      mm: isNamedBy mm:Tchaikovsky-FR-Term , mm:Tchaikovsky-RU-Term ,
      mm:Tchaikovsky-EN-Term .
mm:user_73
      a       mm:PhysicalPerson ;
      mm:isProficient mm:speakEnglish , mm:speakFrench .

// ==== SKILLS ===== //
mm:speakFrench
      a       mm:LinguisticSkill ;
      mm:enablesUnderstandingOf
              mm:French-NL .
mm:speakEnglish
      a       mm:LinguisticSkill ;
      mm:enablesUnderstandingOf
              mm:English-NL .
```

6 Conclusion

In this article, we have presented a conceptual model which enables us to characterize a given denomination's scope of use. We use contextual parameters (job, organization, location) as well as personal parameters (skill, skill level) to make a selection of comprehensible information for a given user. This approach is extendable to any ontological element (class, property, instance) and to other kind of information (such as documents).

We have first shown the relevancy of this approach in the context of collaborative TV production, where amateur contributions to professional workflows are bound to widespread. As the ability to exchange and repurpose information and content becomes a major challenge for organizations, there is a strong need for a system and conceptual model to support them. Models we have studied usually focus on the ability to exchange and transform data between various formats and neglect language variations issues. From a basic distinction between concept, sign and physical carrier we have defined a model which enables us to represent the language variations of user communities and find a way to ease information search and interpretation.

Acknowledgement. The Eureka framework is funding the MediaMap project in which our work takes place.

References

1. Abel, M., Leblanc, A.: Knowledge Sharing via the E-MEMORAe2.0 Platform. In: 6th International Conference on Intellectual Capital, Knowledge Management & Organisational Learning, Montreal, Canada, pp. 10–19 (2009)
2. Rayers, D.J.: Metadata in TV Production: Associating the TV Production Process with relevant Technologies. IBC White Paper, BBC R&D UK (2003)
3. Devlin, B.: What is MXF? Ebu Technical Review, 1–7 (2002)
4. Moulin, C., Sugawara, K., Fujita, S., Wouters, L., Manabe, Y.: Multilingual Collaborative Design Support System. In: 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009), Santiago, Chile, pp. 312–318 (2009)
5. Consultative Committee for Space Data Systems (CCSDS): Reference Model for an Open Archival Information System (2002)
6. Bruno, B.: L'archive numérique: entre authenticité et interpretabilité. archivistes.qc.ca (2000)
7. Aalberg, T., Barthélémy, J., Boutard, G., Görz, G., Iorizzo, D., Jacob, M., et al.: Definition of Object-Oriented FRBR. CIDOC (2008)
8. Summers, E., Isaac, A., Redding, C., Krech, D.: LCSH, SKOS and Linked Data. CoRR (2008)
9. Spyns, P., Meersman, R., Jarrar, M.: Data Modeling Versus Ontology Engineering. SIGMOD Record 31(4), 12–17 (2002)
10. Guarino, N.: Formal Ontologies and Information Systems. In: Proceedings of FOIS 1998, pp. 3–15. IOS Press, Amsterdam (1998)
11. Garcia, R., Celma, O.: Semantic integration and retrieval of multimedia metadata. In: 5th International Workshop on Knowledge Markup and Semantic Annotation, Ireland (2005)

A Specification-Oriented Geospatial Coverage Ontology Study

Jinsongdi Yu, Xia Wang, and Peter Baumann

Jacobs University Bremen,
Campus Ring 1, 28759 Bremen, Germany
{j.yu,xia.wang,p.baumann}@jacobs-university.de

Abstract. In earth sciences, the term “geospatial coverage” denotes a space/time-varying geographic object, that is: multi-dimensional raster data such as 2-D satellite images, 3-D or 4-D seismic data. Web coverage services (WCS) as standardized by the Open Geospatial Consortium (OGC) provide access to such detailed and high-volume sets of information being used in a wide variety of earth system applications, such as in solar, atmosphere, oceans, cryosphere, solid earth, and biosphere research. However, different service implementations sometimes turn out to be incompatible because of incompleteness and a lack of conciseness in the current WCS 1.1.2 standard. The main reason for this is that the specification is written without formal testing procedures. To overcome this, we propose a specification-oriented Ontology for coverages. This work is a part of the development of the new WCS 2.0 standard. In this way, an improved sharing of geospatial coverage data is expected in future.

Keywords: WCS geospatial coverage, raster images, Ontology, OGC WCS.

1 Introduction

Geo data classically are subdivided into vector, raster, and metadata. Vectorial data consist of points, lines, and polygon-bounded areas denoting, e.g., the points of interest, roads, and water areas. Raster data represent regularly spaced observations aligned on a spatio-temporal grid; examples include hyperspectral (i.e., multi-channel) airborne and satellite images, image time series, elevation and bathymetry data, and scanned maps. Metadata encompasses any kind of alphanumeric data attaching further description to the geographic elements, such as parcel ownership, type of road, or date when an image was taken.

To allow for reasoning, ontologies have been modeled on the metadata level as since some time [1][2][3]. To a lesser extent but conceptually convincing, the same has been done with vector data. Raster data, however, currently are not considered and, hence, no inference on the pixel contents is possible at all. Notably, reasoning on raster data does not necessarily mean image understanding. For applications in e-government as well as e-science a pixel-based semantics is used regularly to aid in geo decision making. For example, geo raster users are frequently interested in classification tasks like cloud, snow, and vegetation coverage. Often there is a well-accepted analytical definition for doing such derivations; the Normalized Difference

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 63–74, 2010.

Vegetation Index (NDVI) is evaluated for each pixel of the red and near-infrared (NIR) channel of an earth observation image *a* based on the following formula:

$$NDVI(a) = (a.red - a.nir) / (a.red + a.nir)$$

The result is an image of the same resolution containing pixel values between -1 and +1. The closer a value comes to +1 one, the more likely the pixel location contains vegetation.

The Open GeoSpatial Consortium[1] (OGC) Web Coverage Service (WCS) Standard defines an interface to give clients access to such data in a platform and vendor independent manner. Functionality is organized in a modular manner and ranges from simple access operations (e.g., subsetting) in the core to an open-ended raster query language [4]. In this way, WCS gives access to the original data suitable for further client-side processing, rather than rendering images which are suitable only for display, as it is done, e.g., with Google Maps and OGC WMS.

Obviously, a well-defined data model is of critical importance to achieve true interoperability. The term "coverage" is defined in [5][2] as a space-time varying phenomenon represented by a function which maps points within the space/time extent of the coverage to values, such as spectral intensity or altitude. Raster data are a special case of coverages; further coverage types include, for example, triangulated irregular networks. In Section 3 we will give a detail view on coverages.

Definitions in [5] give only an abstract model, which is refined by WCS to become practically usable. Not always was the coverage model sufficiently well defined. For example, while WCS 1.0.0 [6] elaborated extensively on the metadata, there was no specification of the image contents to be returned – a WCS 1.0.0 server which always delivers a black image is perfectly conformant to this standard version. In its current version WCS 1.1.2 [7], an informal evaluation model is given which, however, does not allow for conformance testing. In the coverage definition of the new version WCS 2.0 [4][8], a formal model of both coverage model and access functions is provided which allows for automated testing.

In this paper, we present an ontology-based modeling of WCS 2.0 coverages which allows to do inference. Ultimately, this will pave the way for including coverage data into the overall orchestration of semantic services. As a short-term goal, we investigate the automated conformance testing to study reasoning on coverages.

Consider the following simple cutout retrieval. Assume that a request is sent to a WCS2.0 service entry for obtaining a subset area with corner points (50,100) and (500,400) from a 2-D raster image of interest (a false color image), see Fig. 1.

The result is a WCS coverage [8] in some image encoding format. The question remains as to how a machine can digest this result and verify whether the response coverage is correct.

The current OGC Compliance & Interoperability Testing & Evaluation[3] (CITE) program uses OGC's Test, Evaluation, and Measurement (TEAM) Engine[4], which is fed with Compliance Test Language [9] (CTL) to digest the geospatial information

[1] http://www.opengeospatial.org
[2] This is a joint work between the OGC and ISO.
[3] http://cite.opengeospatial.org/cite
[4] http://cite.opengeospatial.org/teamengine/

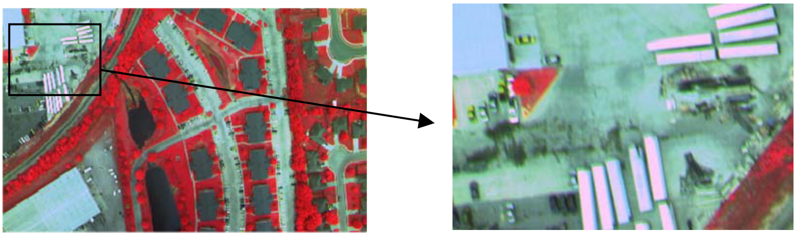


Fig. 1. False color image of a city (left) and cutout obtained from this image (right) (source: http://www.earthlook.org/)

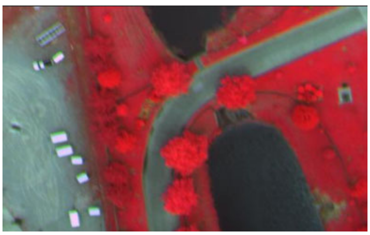


Fig. 2. a. Result with different area

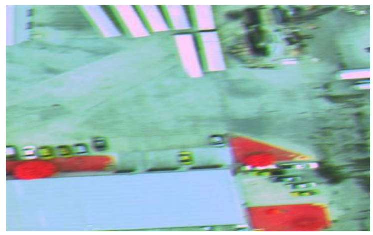


Fig. 2. b. Result with rotated coordinate system

Fig .2. c. Result with wrong pixel intensities

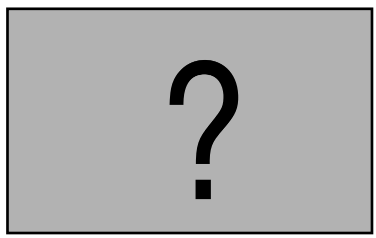


Fig. 2. d. Otherwise wrong result

provided. The CTL uses the eXtensible Stylesheet Language[5] (XSL) to probe the geospatial XML.

However, this approach is not aware of the underlying geospatial semantic. Therefore, it will accept a pseudo-specification compliance implementation. For example, the above retrieval image is just about a 2-D image of the same size as request, but in a different spatial area (see Fig.2.a), rotated coordinate system (see Fig.2.b), false pixel values (see Fig.2.c) or other unknown image result (see Fig.2.d). Without a semantic guarantee, these results will be accepted by a machine.

The specification conformance test needs to check the response GML coverage not only from a syntax level, but also from a semantic level. The geospatial coverage ontology is a formal and machine-understandable representation to address the above mentioned issues. The remaining parts of this paper are organized as follows: Section 2 introduces the related research work, section 3 provides the formalized geospatial

[5] http://www.w3.org/TR/1999/REC-xslt-19991116

coverage definition, section 4 addresses the ontology implementation, and finally, section 5 summaries this paper and discusses the future work.

2 Related Work

There are researches on the geo service semantic to guarantee the information retrieval, such as [10] and [11]. However, further research at geospatial information semantic level is needed to make sure that the retrieval information is correct.

[12] formalized the Conceptual Spaces [13] (CS) model, and used it to bridge between the real-world and symbolic representations. However, such a CS model provides a means to represent the knowledge in geometrical vector spaces, but it does not provide the geospatial coverage knowledge.

[5] describes the coverage knowledge in a semi-formal way, and this needs further development before machines can directly reason on that. XML-schema-based GML serves as a modeling language to express geographical phenomenon, but current GML3.2.1 [17] only provides a conformant, partial implementation of the coverage knowledge.

The existing related ontologies, such as Cyc Open Geospatial Consortium, Cyc Terrain ontology and Drexel ontology only provide some partially related knowledge [14]. Cyc ontologies have not covered some specific concepts, such as coverage domain, range and function [5]. By comparison, the initial aim of this research is to support geospatial semantic digestion in the scope of specification-based geospatial web services formal testing procedures. Drexel developed an OWL representation of a subset of GML 3.0 [15], which is specialized for encoding places of specific types, whereas this geo ontology needs to encode places that occur at a more detailed level, ranging from standardized ISO coverage knowledge to specific WCS coverage knowledge.

Moreover, OGC Web Coverage Processing Service (WCPS) [16] provides a set of so-called probing functions to extract the specified aspects from a coverage. The extraction of these specified aspects relies on the coverage model as defined in [5]; for example, the WCPS probing function imageCrsDomain(*C*) delivers the domain extent expressed in Image Coordinate Reference System (CRS) [18] of a coverage with the identifier *C*.

3 WCS Coverage Ontology

A coverage is formally defined as a function $f: D \rightarrow R$ where D, the coverage's *domain*, represents the spatiotemporal extent in which coverage values can be queried and R, the coverage's *range*, specifies the values which can be associated with coverage locations [5]. A concrete, implementable model of this abstract model is available in the GML 3.2.1 standard [17] where coverages are represented by XML structures defined through XML Schema. This concrete coverage model has been slightly amended by the GML 3.2.1 Application Schema for WCS 2.0 [8], which forms the basis of our discussion here.

In our coverage Ontology, the *AbstractCoverage* is a general concept to define all the shared abstract features of any coverage types. As its subclass, *GridCoverage,* defines raster data. *GridCoverage* is a coverage which has equidistant grid points, but no positioning ("geo reference") in a space-time environment, so that it only can be addressed through its array index coordinates). While *RectifiedGridCoverage* and *ReferenceableGridCoverage*, which are geo rectified or referenced, can be addressed by using spatial-temporal coordinates. The overview of the coverage Ontology is shown in the Tab. 1.

Table 1. Specification-oriented coverage ontology

Abstract Coverage[6]	Domain	Function	Range
GridCoverage	*GridEnvelope*	{*Coverage-Map_pingRule, GridFuction*}	*RangeSet* ⊔ *RangeType*
RectifiedGridCoverage	*GridEnvelope* ⊔ *OffsetVector* ⊔ *Origin*		
ReferencableGridCoverage	*GridEnvelope* ⊔ *CoordTransform*		

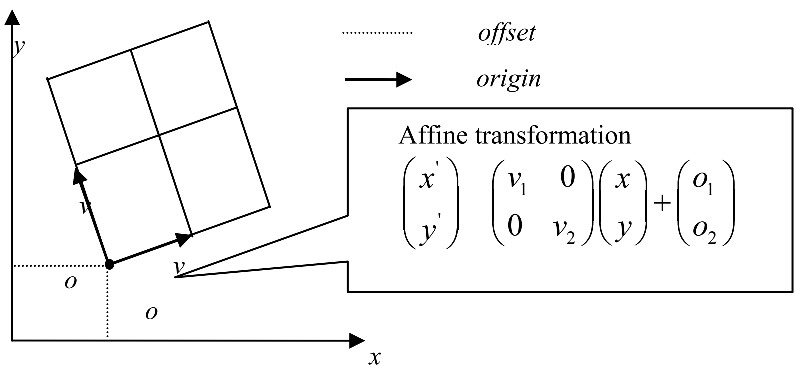


Fig. 3. Geometrical positioning of *RectifiedGridCoverage*

Mathematically, *RectifiedGridCoverage* anchors the array grid to an external spatial-temporal CRS via an affine transformation[7] as described by Fig. 3, while *ReferencableGridCoverage* anchors the grid other than an affine transformation.

The spatio-temporal region, within which the coverage is defined, is described by the coverage *domainSet* [5]. For example, a *GridEnvelope* is used to indicate the *low* and *high* boundary of grid index position, additional *Origin* and *OffsetVector* are used

[6] The research is aimed to address the formal description requirements of multi-dimensional raster data. The other ISO coverage type, such as continuous coverage is not addressed in this research.

[7] http://mathworld.wolfram.com/AffineTransformation.html

to indicate the region of *RectifiedGridCoverage*(see Fig.3), and *CoordTransform* for *ReferencableGridCoverage.*

The set of feature attribute values, within the spatio-temporal region, is described by coverage *rangeSet* [5]. However, current GML3.2.1 coverage lacks of detailed implementation on the range information description. WCS2.0 introduces *rangeType*[8] as well to describe its range structure and meaning, including a name, a data type definition and an optional unit of measure (UOM). The range is delivered in different encoding representations [17], including *AbstractScalarValueList* for scalar value list, *ValueArray* for homogeneous arrays of primitive and aggregate values, *DataBlock* for a block of text encoded values similar to a Common Separated Value (CSV) representation, and *File* for arbitrary external encoding. The external encoding data formats, can be a data format with geospatial descriptions, such as GeoTIFF, NetCDF, and HDF5, or a data format without geospatial descriptions, such as PNG, TIFF, JPEG, and GIF.

The coverage *function* [5] maps the *domainSet* to the *rangeSet* of the coverage. The function can be either a *GridFunction* or a *CoverageMappingRule* [17]. A *GridFunction* includes a *startPoint* for the start index position of a point in the grid that is mapped to the first value in the range set and a *sequenceRule* [5] that describes how the grid points are ordered for association to the elements of the sequence values. An *axisOrder* [17] provides the incrementation order of the grid axis, regularly expressed by regular expression "[\+\-][1-9][0-9]*", implying that "+1 +2" means the points in the grid are to be traversed from lowest to highest on the first and second axis. Fig. 4 shows examples of linear scanning in a 2-dimensional grid under different *axisOrders*. A *CoverageMappingRule* provides a formal or informal description of the coverage function, and these descriptions depend on the further requirements on grid sequence methods. Therefore, it will be either an inline string or a remote reference for the mapping descriptions.

When the *function* is omitted for a coverage, its *startPoint* is assumed to be the *low* boundary of the grid, the *sequenceRule* is assumed to be linear, and the *axisOrder* is assumed to be "+1 +2".

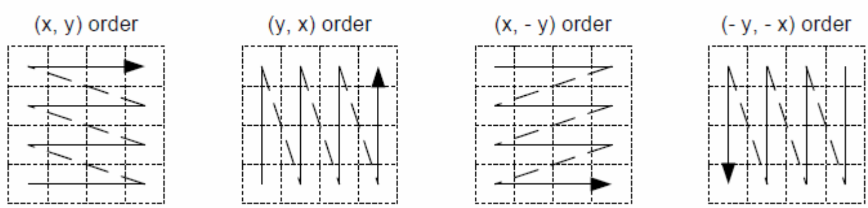


Fig. 4. Examples of linear *sequenceRule* in a 2-dimensional grid [ISO 19123 D.1]

4 Coverage Ontology Implementation

Currently, relevant specifications cannot completely fulfill the requirements of interoperations of heterogeneous geospatial data, and are still under-development or revision

[8] Currently, *rangeType* conforms with the DataRecord definition given in SWE Common [OGC 08-094]

for a future geospatial cloud applications. Basically, our coverage Ontology is defined for digesting the OGC WCS coverage and has been developed with the open world assumption (OWA). It means the coverage-related knowledge, that is not included in or inferred from the knowledge explicitly recorded in this paper, will be considered unknown, rather than wrong or false.

The Web Ontology Language[9] (OWL) is a standard-based language for defining ontologies, based on the Resource Description Framework (RDF and RDF Schema (RDFS) by providing additional vocabulary along with a formal semantics. The behind description logic (DL)[19] can provide provides rich express capabilities for our ontology, specifically, the $\mathcal{SHQ}(\mathcal{D})$ which extends $\mathcal{ALC}$ with transitive role, role hierarchy, nominals and data types. Therefore, we chose OWL as our knowledge representation language and Protégé4.0[10] as our development tool.

The coverage ontology defines a set of axioms which describe the constraints of classes and relationships between concepts and concept properties. The instances of the coverage Ontology is interpreted as a set of individuals with a set of property assertions.

By reusing the coverage example in section 1, we instantialize the returned NIR image and visualize it by OntoGraf[11]. The visualization result can be seen in Fig. 5.

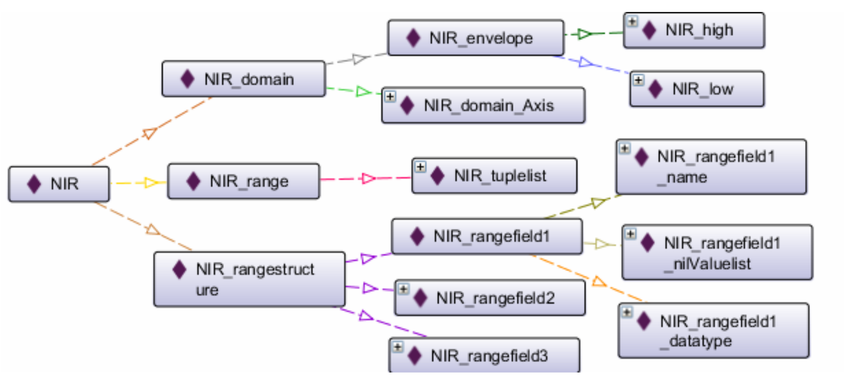


Fig. 5. Visualization of the example WCS coverage ontology

The coverage Ontology defines a key concept, which relies on the ordering list. For example, the coverage boundary *low* and *high* implementations depend on integer list, the *Origin* and *OffsetVector* implementations depend on a double list, and the *AxisOrder* depends on the string pattern implementation. Additionally, the feature attribute *range* relies on the list for different encoding representations.

However, the OWL contains no specific support to ordering support. An approach proposed by [19] was found as a solution to operate the ordering list. We simplified this solution with a recursion pattern, as followings:

$$list = element + list$$

[9] http://www.w3.org/2004/OWL
[10] http://protege.stanford.edu/
[11] http://protegewiki.stanford.edu/wiki/OntoGraf

The visualization result can be seen in Fig. 6.

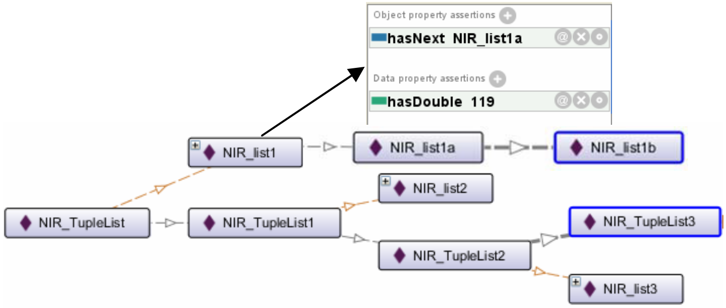


Fig. 6. Visualization of the example coverage's *rangeSet* by ordering list

The response attribute values are represented as a sequential *NIR_TupleListi*, which each one composes of a *NIR_listi* to represent the three band values.

However, practically speaking we found out the solution is computationally expensive and memory consuming when it is applied in the *rangeSet* representation.

An alternative solution is a finite ordered set indexed by non-negative numbers. By this approach, the additional index is added to each list element and the indexing improves the reasoning efficiency of ordering list in the coverage Ontology. The visualization result can be seen in Fig. 7.

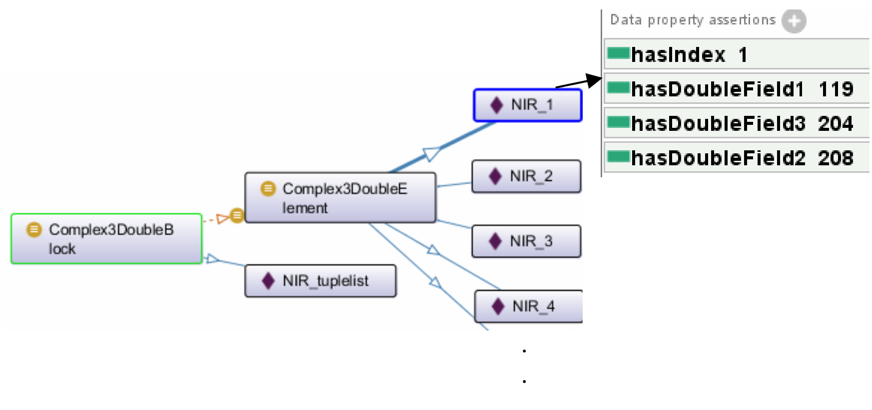


Fig. 7. Visualization of the example coverage's *rangeSet* by ordered set

The response attribute values in *NIR_tuplelist* are represented as a set of complex *NIR_i* elements, which each one composes of an index to indicate the order and three double fields to represent the three band values.

However, this modeling does not ensure that the list is well formed. Further rules (e.g. SWRL rules) are needed to check the integrity of the list representation. For example, the coverage boundary *low* and *high* are constrained by coverage dimension; *rangeSet* is constrained by coverage boundary and range structure.

5 Conclusions and Outlook

Geospatial ontologies are crucial in geospatial data description. Although there are a number of geospatial ontologies, there is still a lack of a suitable coverage ontolgies for WCS coverage services, specifically the WCS2.0 version. Based on the related geo coverage specifications, we modeled the geospatial coverage Ontology. This Ontology can particularly seamlessly describe WCS coverage, and support the OWL reasoning on both metadata and pixel level.

In order to adapt to the dynamic cloud computing environment, it is an evolution process to model such coverage Ontology. It involves ontology enrichment along with adding new concepts, e.g., *ContinuouCoverage*, topology relationship, quality description and so on. It also includes ontology revision, such as the addition or removal of concept mapping.

Acknowledgments. For the work reported in this article we gratefully acknowledge the financial support of the European Space Agency (ESA) through the HMA-FO Online Data Access project and the China Scholarship Council (CSC) through stipends, as well as Earth Science System Research School (ESSReS). The authors would like to thank Katrina Stollmann for fruitful academic discussions.

References

1. Parekh, V., et al.: Ontology based Semantic Metadata for Geoscience Data. In: Proceedings of the International Conference on Information and Knowledge Engineering, IKE 2004, pp. 485–490. CSREA Press, Las Vegas (2004)
2. Jia, Q., Guo, S.Z., Chen, H.Y., Li, N.: Research on Ontology-Based Metadata Model. In: First International Workshop on Database Technology and Applications, pp. 363–366. IEEE press, Wuhan (2009)
3. Mitschick, A., Meißner, K.: Generation and Maintenance of Semantic Metadata for Personal Multimedia Document Management. In: 1st International Conference on Advances in Multimedia, pp. 74–79. IEEE Press, Colmar (2009)
4. Baumann, P. (ed.): WCS 2.0 Overview: Core and Extensions. Number 09-153. OGC (2009)
5. n.n.: Geographic information—coverage geometry and functions. Number 19123:2005. ISO (2005)
6. Evans, J.D. (ed.): Web Coverage Service (WCS), Version 1.0.0. Number 03-065r6. OGC (2003)
7. Whiteside, A., Evans, J.D. (eds.): Web Coverage Service (WCS) Implementation Standard, Version 1.1.2. Number 07-067r5. OGC (2007)
8. Baumann, P. (ed.): GML 3.2 Application Schema for WCS 2.0. Number 19-146. OGC (2009)
9. Morris, C. (ed.): Compliance Test Language (CTL) Best Practice. Number OGC 06-126r2. OGC (2006)
10. Di, L., Zhao, P., Yang, W., et al.: Ontology-driven Automatic Geospatial-Processing Modeling Based on Web-service Chaining. In: Proceedings of the Sixth Annual NASA Earth Science Technology Conference. College Park, MD, USA (2006)

11. Lan, G., Huang, Q.: Ontology-based Method for Geospatial Web Services Discovery. In: Int. 2007 Conference on Intelligent Systems and Knowledge Engineering (2007)
12. Dietze, S., Domingue, J.: Enriching Service Semantics through Conceptual Vector Spaces. In: Workshop: Workshop on Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science (ONTOSE 2009), at: The 21st International Conference on Advanced Information Systems (CAiSE 2009), Amsterdam, NL (2009)
13. Gärdenfors, P.: Conceptual Spaces - The Geometry of Thought. MIT Press, Cambridge (2000)
14. Ressler1, J., Dean, M.: Geospatial Ontology Trade Study. In: OIC 2007, Columbia, Maryland, pp. 55–59 (2007)
15. Cox, S., Daisey, P., Lake, R., et al. (eds.): OpenGIS Geography Markup Language (GML) Encoding Standard, Version 3.0. Number 02-023. OGC (2002)
16. Baumann, P. (ed.): OpenGIS Web Coverage Processing Service (WCPS) Language Interface Standard, Number OGC 08-068r2. OGC (2008)
17. Portele, C. (ed.): OpenGIS Geography Markup Language (GML) Encoding Standard, Version 3.2.1. Number 07-036. OGC (2007)
18. Lott, R. (ed.): OGC Abstract Specification Topic 2, Spatial referencing by coordinates, Number 04-046r3. OGC (2004)
19. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, Applications. Cambridge University Press, Cambridge (2003)
20. Drummond, N., Rector, A.L., Stevens, R., Moulton, G., Horridge, M., Wang, H., Seidenberg, J.: Putting OWL in Order: Patterns for Sequences in OWL. In: 2nd OWL Experiences and Directions Workshop, Athens, GA (2006)

Appendix: The DL Definitions

*definition*1

$$AbstractCoverage \doteq \forall hasDomainSet.DomainSet \sqcap = 1 hasDomainSet$$
$$\sqcap \forall hasFuntion.fuction \sqcap \leq 1 hasFunction$$
$$\sqcap \forall hasRangeSet.rangeSet \sqcap = 1 hasRangeSet$$
$$\sqcap \forall hasRangeStructure.rangeType \sqcap = 1 hasRangeStructure$$

$$GridCoverage \sqsubseteq AbstractCoverage$$

$$RectifiedGridCoverage \sqsubseteq AbstractCoverage \sqcap \neg GridCoverage \sqcap \neg ReferencableGridCoverage$$

$$ReferencableGridCoverage \sqsubseteq AbstractCoverage \sqcap \neg GridCoverage \sqcap \neg RectifiedGridCoverage$$

*definition*2

$$GridCoverage \doteq \forall hasDomainSet.Grid \sqcap = 1 hasDomainSet$$
$$\sqcap \forall hasFuntion.fuction \sqcap \leq 1 hasFunction$$
$$\sqcap \forall hasRangeSet.rangeSet \sqcap = 1 hasRangeSet$$
$$\sqcap \forall hasRangeStructure.rangeType \sqcap = 1 hasRangeStructure$$

*definition*3

$$RectifiedGridCoverage \doteq \forall hasDomainSet.RectifiedGrid \sqcap = 1 hasDomainSet$$
$$\sqcap \forall hasFuntion.Fuction \sqcap \leq 1 hasFunction$$
$$\sqcap \forall hasRangeSet.RangeSet = 1 hasRangeSet$$
$$\sqcap \forall hasRangeStructure.rangeType \sqcap = 1 hasRangeStructure$$

*definition*4

$$ReferencableGridCoverage \doteq \forall hasDomainSet.ReferencableGrid \sqcap = 1 hasDomainSet$$
$$\sqcap \forall hasFuntion.fuction \sqcap \leq 1 hasFunction$$
$$\sqcap \forall hasRangeSet.rangeSet = 1 hasRangeSet$$
$$\sqcap \forall hasRangeStructure.rangeType \sqcap = 1 hasRangeStructure$$

*definition*5

$$GridEnvelope \doteq \forall hasLow.Integer \sqcap = 1 hasLow \sqcap \forall hasHigh.Integer \sqcap = 1 hasHigh$$

*definition*6

$Grid \sqsubseteq DomainSet$

$RectifiedGrid \sqsubseteq DomainSet \sqcap \neg Grid \sqcap \neg ReferencableGrid$

$ReferencableGrid \sqsubseteq DomainSet \sqcap \neg GridCoverageDomain \sqcap \neg RectifiedGridCoverageDomain$

*definition*7

$Grid \doteq DomainSet \sqcap \forall hasGridEnvelope.GridEnvelope \sqcap =1 hasGridEnvelope$

*definition*8

$$RectifiedGrid \doteq DomainSet \sqcap \forall hasGridEnvelope.GridEnvelope \sqcap =1 hasGridEnvelope$$
$$\sqcap hasOrigin.Origin \sqcap =1 hasOrigin$$
$$\sqcap hasOffsetVector.OffsetVector \sqcap =1 hasOffsetVector$$

*definition*9

$$ReferencableGrid \doteq DomainSet \sqcap \forall hasGridEnvelope.GridEnvelope \sqcap =1 hasGridEnvelope$$
$$\sqcap hasCoordTransform.CoordTransform =1 hasCoordTransform$$

*definition*10

$AbstractScalarValueList \subseteq RangeSet$

$ValueArray \sqsubseteq RangeSet \sqcap \neg AbstractScalarValueList \sqcap \neg DataBlock \sqcap \neg File$

$DataBlock \sqsubseteq RangeSet \sqcap \neg AbstractScalarValueList \sqcap \neg ValueArray \sqcap \neg File$

$File \sqsubseteq RangeSet \sqcap \neg AbstractScalarValueList \sqcap \neg DataBlock \sqcap \neg ValueArray$

*definition*11

$GridFuction \sqsubseteq Function$

$CoverageMappingRule \sqsubseteq Function \sqcap \neg GridFuction$

*definition*12

$$GridFuction \doteq \forall hasStartPoint\ StartPoint \sqcap \leq 1 hasStartPoint$$
$$\sqcap \forall hasSequenceRule\ SequenceRule \sqcap \leq 1 hasSequenceRule$$
$$\sqcap \forall hasAxisOrder.AxisOrder \sqcap \leq 1 hasAxisOrder$$

*definition*13

$$CoverageMappingRule \doteq \left(\forall hasRuleDefinition\ String \sqcap \leq 1 hasRuleDefinition\right)$$
$$\sqcup \left(\forall hasRuleReference.anyURI \sqcap \leq hasRuleReference\right)$$

A Space-Based Interoperability Model

Diego Bernini, Daniela Micucci, and Francesco Tisato

Department of Informatics, Systems and Communication,
University of Milano - Bicocca,
Viale Sarca 336, 20126 Milan, Italy
{diego.bernini,daniela.micucci,francesco.tisato}@disco.unimib.it
http://www.sal.disco.unimib.it

Abstract. Software components may need to reason on thematic information that are usually contextualized in several and different dimensions. A person, for instance, may be physically located in a room; may play a specific role in an organization chart; and may be associated with an RFID tag in a set of identifiers. Moreover, software components may require thematic information which are produced by other software components and, in turn, they may be sources of information. This paper presents an interoperability model based on spatial concepts. The spatial model supports the contextualization of thematic information in different spaces - be they conceptual or physical. Main concepts are spaces, locations and mapping between locations belonging to different spaces. The model constitutes the basis to enable information sharing between software components. From this point of view, software components declare their interest in information from specific spatial locations (subscription contexts) and diffuse information in possibly other specific spatial locations (publication contexts). In a framework reifying the model we propose, software components may retrieve information without knowing who are the other components in the system. Indeed, the model enforces the indirect communication between software components which may rely on different views of the information domain.

Keywords: space-awareness, localized information, interoperability, complex and distributed system.

1 Introduction

Ubiquitous computing, pervasive computing, ambient ecologies, ambient intelligence, smart spaces, responsive environments, and so forth, are nowadays emerging interaction models. Their applications range from automated monitoring and control of homes for safety and comfort in Domotics [1] to Interactive Architecture, Design and Art [2].

Realizing large-scale systems that reify interaction models requires developing software systems that integrate various heterogeneous devices and technologies for sensing different properties of the environment and for providing different kinds of services. Typical input components are motion sensors, localization systems, cameras and vision systems; while common output ones are screens, lights,

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 75–89, 2010.

audio synthesizers and general electronic appliances. In such systems interoperability is a key requirement.

A successful architecture should model interoperability so that software components may be not aware about the existence of the other software components. Indeed, such an architecture allows to "indirectly connect" software components even if they are heterogeneous.

The devised interoperability model should be then realized by an underlying framework. To be general, that is, to support any kind of system, such a framework must rely on a generic "communication model" that is, it must not depend on a specific application domain. In other words, the framework should be aware of a few basic concepts that may be considered general enough to be applicable in several application domains. This requires, at least, that the semantics of the information is left to applications. Moreover, the domain of information should be arranged in an appropriate way that, in turn, must be general enough to be applied in several domains.

The paper presents a space-based model supporting interoperability among software components. The model we present aims at identifying the services and the mechanisms that a framework should support to guarantee cooperation in terms of information sharing between software components.

The model exploits space as the metaphor to model the domain of information and derives from space-awareness the interoperability model.

Space-based concepts are suitable to represent both physical spatial models (e.g., geo-referred) and conceptual spatial models (e.g., organization structure). Software components exploit suitable spaces - which are defined according to generic spatial models - to indirectly interact with other software components.

Thematic information is localized in several spaces. Talking about localized information suggests a clean separation between spatial contextualization ("where") and thematic information ("what"), whose meaning depends on software component interpretation beyond its localization. Spaces and localized information constitute the domain of information of the system.

Likewise the semantics of information is left to applications, the semantics of spaces is, in turn, up to applications only. This ensures that the framework reifying the model is generic to support any kind of application domain in which the generic spatial model suffices to model the domain of information.

Obviously, in order to cooperate, software components must share a common view of the domain of information, or, in other terms, they must rely on an ontology of the the domain of information.

Unfortunately, software components usually rely on a view of the domain of information that is subjective (i.e., partial) and, what is worse, they do not rely on the same view.

A mapping is then required to harmonize different subjective views. A mapping enables an information localized by a software component in its subjective view to be accessed by another software component relying on a different subjective view possibly derived by different spatial models.

Finally, the space-based interoperability model must provide the basic space-based concepts enabling a software components both to access the information they need (i.e., they specify a subscription context) and to diffuse the information they produce (i.e., they publish a thematic information within a publication context).

The space model supporting interoperability is built over a revised version presented in [3] and [4]. [3] introduces a common set of space-related concepts that provides applications with a unique view of the spaces. The devised model allows software components to rely on an operational view of the space, which is defined in terms of opaque locations and of movements between them. Actual space model maintains the opaqueness of locations, but discards movements assuming that software components are aware about the structure of the space. [4] presents a space-aware architecture based on the space model described in [3].

The rest of the paper is organized as follows. Section 2 describes spatial concepts that are the basis of the interoperability model described in Section 3. Spatial concepts and the interoperability model are presented by means of UML [5] class diagrams enriched with OCL [6] constraints when required. Section 4 presents some related works. Finally, Section 5 sketches some conclusions and discusses future developments.

2 Spatial Concepts

Space-based concepts are suitable to represent both physical spatial models (e.g., geo-referred) and conceptual spatial models (e.g., organization structure). Software components exploit suitable spaces - which are defined according to generic spatial models - to indirectly interact with other software components.

2.1 Generic Spatial Models

A *generic spatial model* define a space typology. A *space* is an instance of a spatial model and it is a collection of *location*s. The related spatial model specifies how locations are defined.

In a space there cannot be two locations that are equal. Moreover, each location is part of one space only.

Thematic information may be localized in one or more locations belonging to one or more spaces.

Different spatial models may be identified. In this version we deal with finite spaces. The graph spatial model supports the definition of graph spaces whose locations are nodes and edges. The grid spatial model supports the definition of spaces whose locations are arranged in a bi-dimensional grid and are characterized by a pair of indexes. Finally, a name spatial model just supports the definition of name spaces as collections of named locations.

Fig. 1 sketches the UML class diagram modelling generic spatial models.

The abstract class `Location` reifies a generic location. It is characterized by an attribute - called `position` - specifying its position in a space.

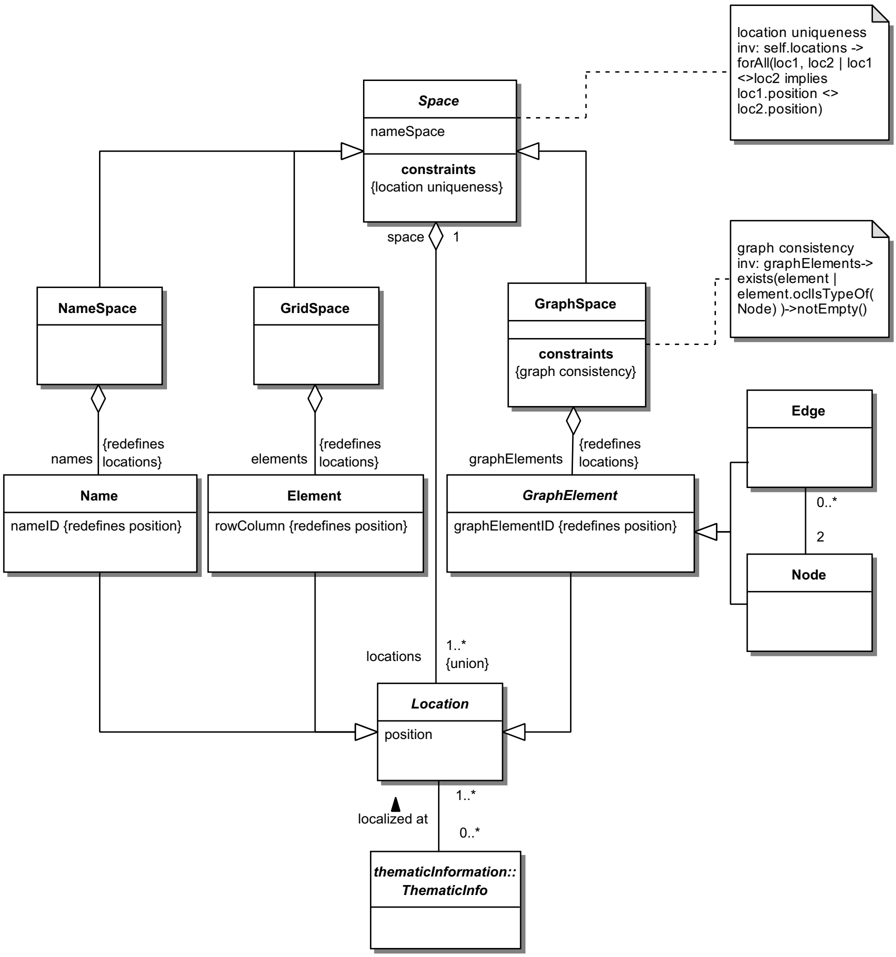


Fig. 1. Spatial Models

The abstract class `Space` reifies a generic spatial model. The aggregation with the `Location` class specifies that a space is a collection of locations. In a space there must exist at least one location. This is specified by the `1..*` cardinality in the association-end (the one with the role labelled `locations`) of the aggregation. Moreover, a location belongs to one space only. This is expressed by the `1` cardinality in the association-begin (the one with the role labelled `space`) of the aggregation. Since aggregations specify a Set type collection (i.e., a collection in which duplicates are not allowed), it does not ensure uniqueness in terms of instances state too. In the specific case, this means that even if two locations belonging to the same space are different instances, they may have the same value for `position`. To overcome the drawback, we introduce the constraint *location uniqueness*. Obviously, `Location` is an abstract class, but defining this property at this level allows generalizing the constraint that must hold for each derived class. As we will see, the uniqueness property is fundamental for localizing and sharing thematic information.

From the more general definition of spatial model (`Space`), we derived the following specific spatial models: `GraphSpace` (i.e., a space representing a graph), `GridSpace` (i.e., a space representing a bi-dimensional grid), and `NameSpace` (i.e., a space representing a collection of names). Obviously, other typologies may be identified.

The derived spatial models redefines both `Location` and `position` to accomplish their spatial characteristics.

Concerning with `GraphSpace`, we introduced the *graph consistency* constraint stating that if a graph is defined by one element only, such element must be a `Node`.

Finally, `ThematicInfo` represents any kind of spatially-localized information. An information is spatially-localized when it is bound to one or more locations. Such locations my be part of different spaces.

Generic spatial models are defined once and for all. Their are generic enough to model a set of commonly used spaces and, since are generic, they may be used in several application domains. Indeed, applications using such spatial models should provide them the correct semantics.

Software components rely on spaces that are instances of generic spatial models. Such spaces constitute the *awareness* that a software component has about the overall system (i.e., the domain of information). Such awareness is termed *subjective view* of the domain of information since it may not contain all the defined spaces. Different software components may rely on different subjective spaces. Thus, the intersection between two subjective spaces may be the empty set.

2.2 Space Mappings

One or more locations belonging to a space may be related to one or more locations belonging to a different (another) space. When a relation between two locations exists, then a *binary map* is defined. A binary map is a relation with direction: from a source to a target. If two locations need to be mapped in both directions, then two binary maps must be defined.

Mapping helps in making software components able to share information even if they are not aware of the same spaces, that is, they rely on different subjective views. Suppose to have two spaces (*aSpace* and *bSpace*) and a map between *locA* (source) belonging to *aSpace* and *locB* (target) belonging *to bSpace*. If a software component *s1* is aware of *spaceA* and localizes a thematic information *t1* in *locA*, the existing map between *locA* and *locB* indirectly localizes *t1* in *locB* too. If there is another software component *s2* that is aware of *bSpace* and is interested in thematic information localized in *locB*, it will be able to retrieve *t1*.

This way, software components may rely on spaces they are aware without knowing the spaces (and related spatial models) exploited by the other software components.

Fig. 2 sketches spatial concepts related to mapping.

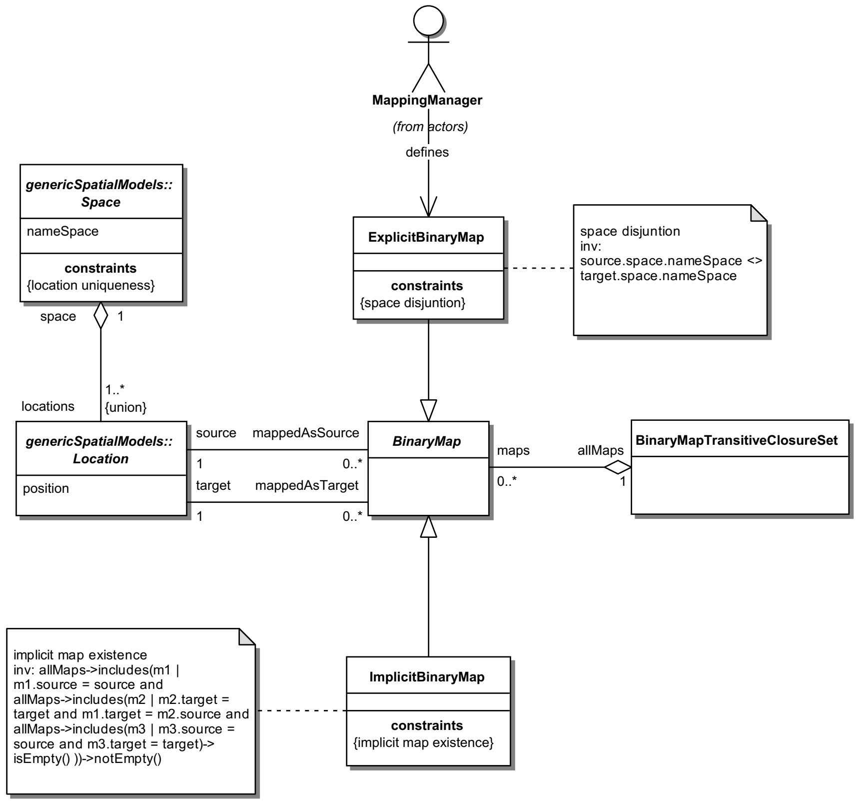


Fig. 2. The Mapping

The `BinaryMap` class specifies a generic mapping. A mapping is a binary relation involving one location acting as `source` and another location acting as `target`. This is explicated by the two associations relating `BinaryMap` and `Location` classes.

`ExplicitBinaryMap` is a kind of binary map that is explicitly defined by associating a source location to a target location. The constraint *space disjunction* states that the source and the target must belong to different spaces.

`ImplicitBinaryMap` is another kind of binary map. It maps a source location to a target one too, but it is implicitly derived from applying the transitive property to the explicit mappings. In particular, it is considered the restricted transitive closure of the explicit mappings. With restricted transitive closure of a mathematical relation R (the set of explicit mappings in our case) we intend the relation T containing all the pairs:

– *(a,b)* belonging to R
– *(a,c)* where *(a,z)* and *(z,c)* belongs to T for some z and c is not equal to a.

The constraint termed *implicit map existence* bounds the existence of an `ImplicitBinaryMap` by a set of properties reifying the transitive closure.

Finally, the `BinaryMapTransitiveClosureSet` maintains all the binary maps.

3 Space-Based Interoperability Model

Spatial concepts presented in the previous section constitute the building blocks of the interoperability model. To share thematic information with other parties, software components localize information in *publication context*s. Parties interested in localized-information specify *subscription context*s. A publication context and a subscription context possibly rely on different spaces. A *match* occurs when the contexts are directly or indirectly related due to the existence of at least a binary map involving a location of the publication context acting as source and a location of the subscription context acting as target. When a match occurs, the software component who made the subscription receives the information and its complete spatial contextualization.

3.1 Contexts and Publications

A *context* specifies a set of locations belonging to the same space. A *publication context* is the context used by a software component when sharing a thematic information. After a contextualization, the information will be localized in all the locations specified in the context. On the other side, a *subscription context* is the context used by a software component when specifying the locations in a space that are relevant to it, that is, locations that may have localized thematic information useful to the software component and generated by other software components.

Fig. 3 sketches the classes modelling contexts. The more general `Context` class specifies a set of locations defined in the same space. It may contain from 1 location only as far as the whole locations in the space.

The constraint *space uniqueness* ensures that all the contained locations belong to the same space.

Two special kinds of `Context` are defined:

- `PublicationContext` is the context specified by a software component acting as information source (`InformationSource`);
- `SubscriptionContext` is the context specified by a software component (`InformationTarget`) acting as information sink.

`PublicationContext` and `SubscriptionContext` only differ in that the former is used by the software component sharing a information, while the latter is used by the software component to subscribe. Really they are simple contexts.

A software components acting as information source contextualizes thematic information by defining a *publication*, that it the pair (publication context, thematic information). In Fig. 4 `Publication` models a publication act. The class is in association with `1..*` (one or more) `PublicationContext` and with one `ThematicInfo`. Publishing an information implies localizing the information in

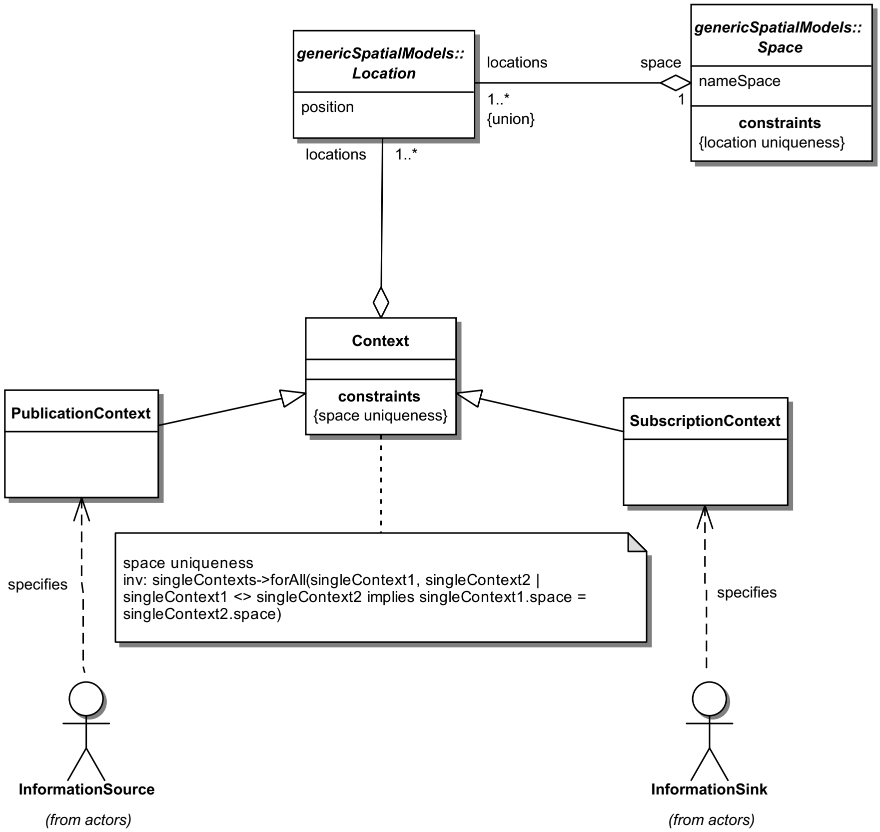


Fig. 3. Contexts

all the locations specified in the publication context and in all the locations belonging to different spaces for which a binary map exists (explicitly or implicitly).

3.2 Matching

Publication context and subscription context possibly rely on different spaces. In any case, information published on a publication context *PC match* a subscription context *SC* if (see Fig. 5):

- *direct match*: the intersection of the sets of locations specified in *SC* and *PC* is not empty, or
- *indirect match*: the set of locations specified in *PC* includes a location *L* which is mapped explicitly or implicitly to a location *L'* belonging to *SC*. Indirect match exploits the restricted transitive closure of the enumerative mappings (as presented in subsection 2.2).

Direct match is modelled by the `DirectMatch` class. The constraint *direct match existence* specifies the condition under which a direct match exists. The condition

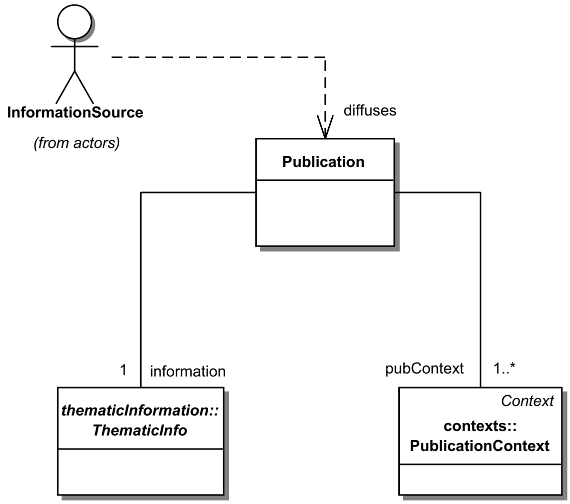


Fig. 4. Publications

is quite trivial: a direct match occurs if publication context and subscription context share a location. If this kind of match occurs, both the software components have respectively published and subscripted in the same space.

Indirect match is modelled by the `IndirectMatch` class which, in turn, has defined the constraint *indirect match existence* that specifies the conditions under which an indirect match exists. This case, the software component who made the subscription has specified a different space from the one used to publish information. The match occurs if there exists at least a location in the publication context that acts as source in a binary map and there exits a location in the subscription context that acts as the target of the binary map. In other words, a (implicit or explicit) map must exist between a location in the publication context and a location in the subscription context.

`Match` class is the superclass of both `DirectMatch` and `IndirectMatch`. It maintains related the `SubscriptionContext` and the `PublicationContext` that have matched (as specified by the two associations with end-roles labelled `subcontext` and `pubContext` respectively).

Note that both the aggregation relating `Space` and `Location` and the constraint defined in `Space` ensure that a publication reaches a subscriber.

3.3 Matched Publications

The result of a match is a *matched publication* that includes the information enriched with the publication contexts where the information was originally published and those which are explicitly or implicitly mapped from them, according to the restricted transitive closure (*context completeness*).

In Fig. 6, a matched publication is modelled by the `MatchedPublication` class. It is in association with the `ThematicInfo` originally published and one

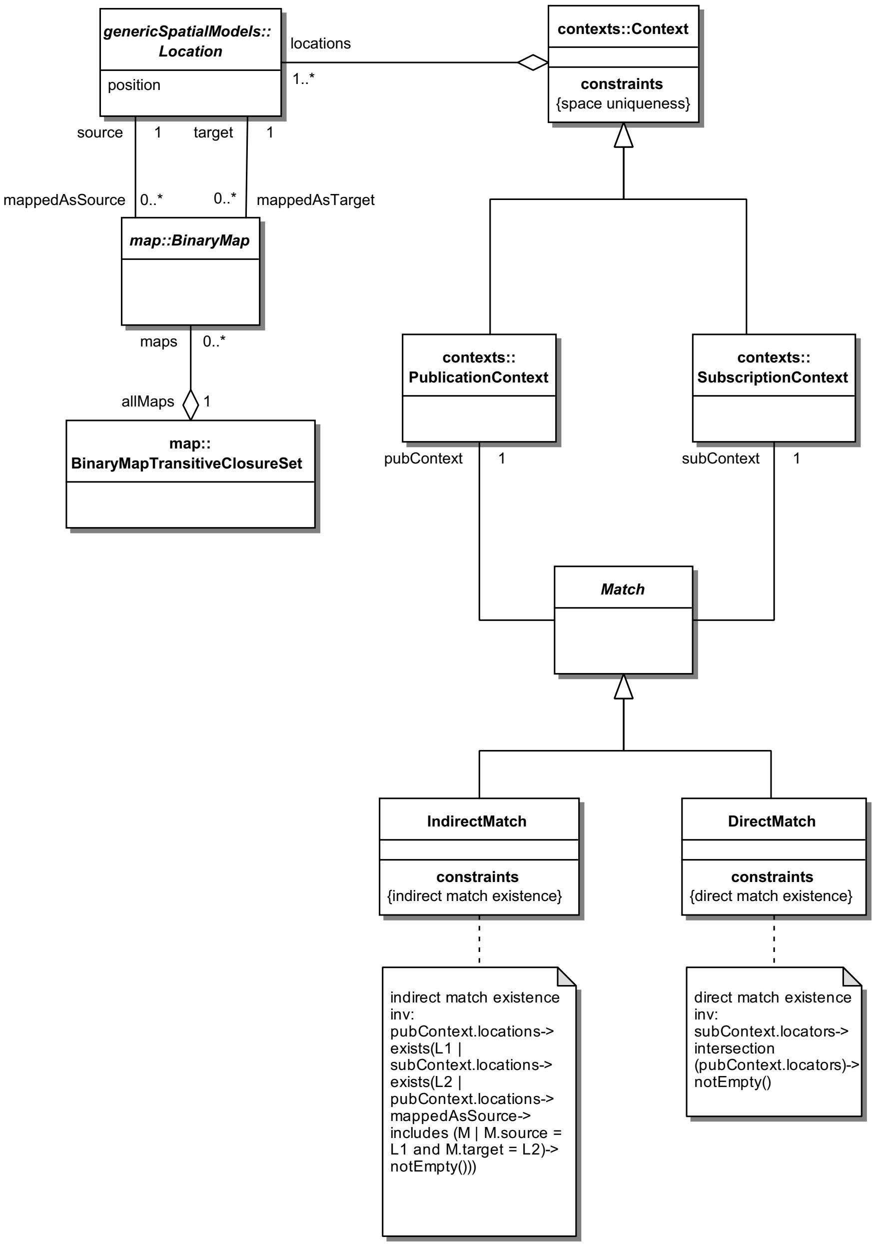


Fig. 5. The match

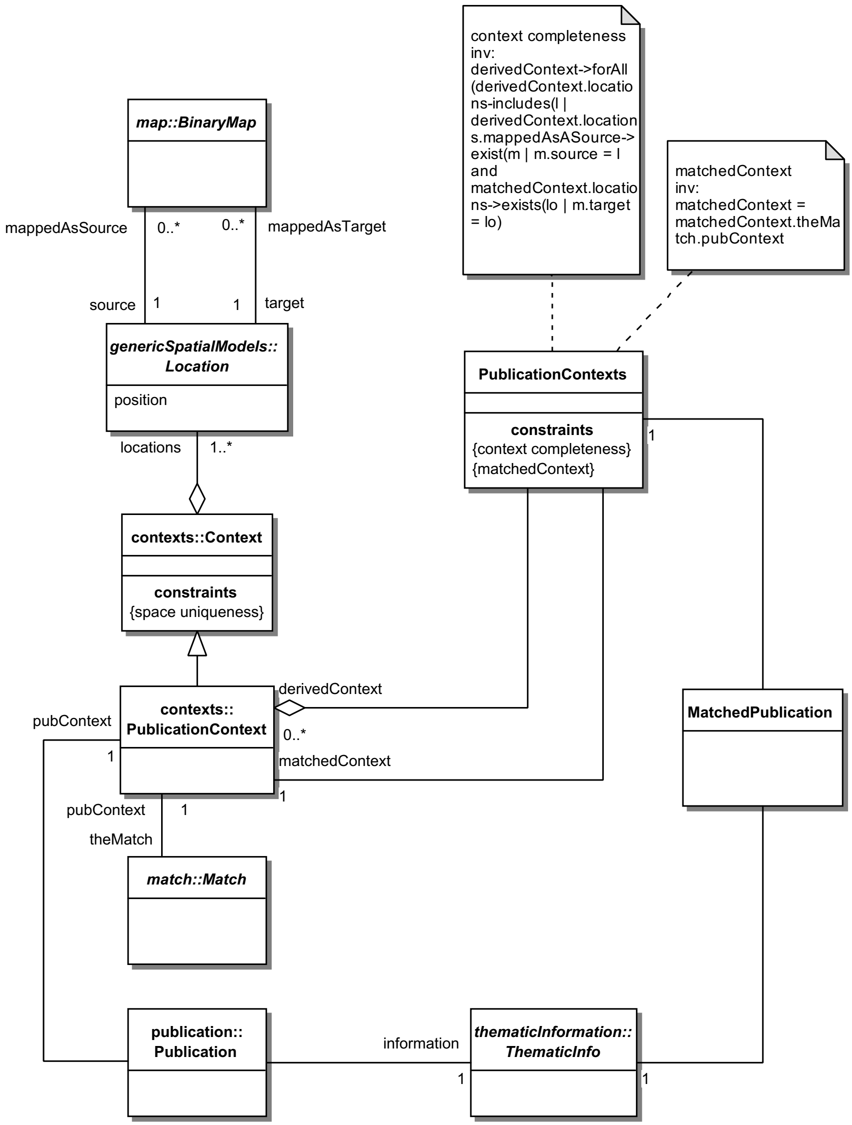


Fig. 6. Matched publication

PublicationContexts. PublicationContexts is a set of PubblicationContext that contains the original publication context (matchedContext) that matched the publication against the subscription and possibly a set of other publication contexts (derivedContext).

derivedContext model all the contexts in which each contained location is the target in a binary map where a location in matchedContext is its source. In other words, the derivedContext reifies the context completeness (constraint *context completeness* in PublicationContexts class).

4 Related Works

The concept of space has been widely studied in computer science as several domains (e.g., emergency management systems, video-surveillance, smart spaces, collaborative and pervasive computing) take advantage from it.

Space models are often based on an ontology that defines the relevant space-related concepts in a non-ambiguous way. For example, in [7] such a set contains: how space is modeled (explicitly or implicitly), the way physical objects are modeled, the spatial relationships which hold for entities located in spaces, and the concept of frames of reference. The first category is relevant to this paper. Space modeling can take one of two basic views: the *Newtonian*, in which the space exists independently of the objects that may be located in it (this view is also called substantival view [8] or general space [9]), and the *Leibnizian* view, in which the space exists as a matter of inter-relationships between objects (this view is also called relational [8] or local space [9]). An exhaustive overview of spatial concepts may be found in [10].

Different ontologies have been proposed. An exhaustive comparison may be found in [11] and [12]; whereas [7] provides their evaluation based on the expressiveness with respect to space-related aspects.

Most of the proposed spatial ontologies carefully separate entities from their location in the space, as claimed in [13], [14] and [7], to obtain what in software engineering is termed separation of concerns [15]. In this line is the ontology proposed in [16], as a generalization of [17]. It is an interesting step towards the separation between entity, space and time. It looks at relationships as first class objects. Specifically, through relationships, information should be related to its thematic aspect, to its spatial location and its occurrence in time: the thematic domain is indirectly associated to the space via the located_at and occurred_at relations and to time since these relations are enriched by discrete time stamps. Such an approach ensures flexibility in integrating thematic and spatial information.

From an software architecture point of view, the approaches proposed to interaction-based systems differ in how the interactions between components are achieved. Following [18] we can identify three main kinds of interaction styles.

In *direct interaction* components interact directly through different kinds of protocols, like message passing and (remote) procedure calls. CORBA [19] and the Open Service Gateway initiative (OSGi) [20] are popular examples of general purpose framework supporting direct interactions. Direct communication can be very efficient, however the main lack of this approach is that components should know each other or at least the interfaces of the other components. This can obstacle the integration of new and heterogeneous components.

Shared data spaces are typical mediums to achieve indirect interaction. Components interact through common data structures. These data structures are hosted in some centralized data space (e.g., a traditional LINDA-like tuple space [21]), or they can be fully distributed over different computational nodes. In this way openness to heterogeneous components is promoted, because interaction is based on an agreement on the data structures. Equip [22] is an example of shared data space conceived for building pervasive applications.

According to the *event-based publish/subscribe interaction style*, components can publish events and react to events of interest that are indicated through subscriptions. Subscription patterns and matching rules define the relation among subscriptions and events of interest. This style allows a strong decoupling between components which can ignore the identities of other components or even their existence. Context-based publish subscribe separates events from their contexts: subscriptions are based on the events' contexts, not on the events themselves. [23] proposes a generic model that introduces the concept of "context of relevance" and "context of interest" which are comparable to our model's contexts.

A framework reifying our model will combines context-based publish subscribes and the shared data spaces interaction styles. Components publish information on publication contexts and they receive information specifying subscription contexts trough subscriptions. The matching rules consider only spatial contexts and their mappings. [18] is also based on the combination of these two styles. It allows to explicitly program the matching rules but it does not strongly separate information from its context.

Finally, differently from other works like [24], a framework supporting our model will not focus on spaces representing physical locations only: information can be located in multiple spaces with different semantics, either physical or logical. We claim that this is a key conceptual aspect in realizing interaction-based systems. In fact it promotes fully context-based interactions, according to the definition of context as "any information that can be used to characterize the situation of entities [...]" [25].

5 Conclusion and Ongoing Activities

The paper presented a model for interoperability based on spatial concepts. The space-related concepts devised are general enough to be applied to several application domains. Such concepts require an underlying framework supporting them and providing a set of services that subscribers and publishers may use to diffuse and receive information.

A reasonable framework supporting our model will combine context-based publish/subscribe and the shared data spaces interaction styles. Components publish information on publication contexts and they receive information specifying subscription contexts trough subscriptions. The matching rules consider only spatial contexts and their mappings.

A prototypal implementation of a framework reifying the presented model has been realized. The platform has been called SIS (Space Integration Services) [26], and is being experimented in the context of the project "GAS (Grandi Attrezzature Scientifiche) - Intelligent Building", University of Milano Bicocca. The project aims at realizing systems in which several and heterogeneous acquisition devices provide information to applications like tracking people, localizing specific persons inside the building and so on.

Finally, we are planning to enrich the spatial models with metrics allowing properties like adjacency, distance between locations, paths and the like to be defined.

Acknowledgments. This work was supported in part by MIUR-FIRB Integrated System for Emergency "InSyEme" project under the grant RBIP063BPH.

References

1. Aiello, M., Dustdar, S.: Are our homes ready for services? a domotic infrastructure based on the web service stack. Pervasive Mob. Comput. 4(4), 506–525 (2008)
2. Bullivant, L.: Responsive Environments: architecture, art and design. Victoria and Albert Museum (2006)
3. Tisato, F., Micucci, D., Adorni, M., Cirasa, E.: Architectural abstractions for space awareness. In: Proceedings of the 2nd International Workshop on Ontology, Conceptualization and Epistemology for Software and Systems Engineering (ONTOSE) (2007)
4. Micucci, D., Tisato, F., Adorni, M.: Engineering spatial concepts. The Knowledge Engineering Review 24(Special Issue 01), 77–93 (2009)
5. Group, O.M.: OMG Unified Modeling LanguageTM (OMG UML), Superstructure (February 2009), `http://www.omg.org/spec/UML/2.2/Superstructure`
6. Group, O.M.: Object Constraint Language (February 2010), `http://www.omg.org/spec/OCL/2.2`
7. Bateman, J., Farrar, S.: Spatial ontology baseline. Technical Report SFB/TR8 internal report I1-[OntoSpace]:D2, Collaborative Research Center for Spatial Cognition, University of Bremen, Germany (2004)
8. Grenon, P.: Tucking RCC in cyc's ontological bed. In: Proceedings of the 18th International Joint Conference on Artificial intelligence, Acapulco, Mexico, pp. 894–899. Morgan Kaufmann Publishers Inc, San Francisco (2003)
9. Vieu, L.: Spatial representation and reasoning in artificial intelligence. In: Spatial and Temporal Reasoning, pp. 5–41 (1997)
10. Habel, C., Eschenbach, C.: Abstract structures in spatial cognition. In: Freksa, C., Jantzen, M., Valk, R. (eds.) Foundations of Computer Science. LNCS, vol. 1337, pp. 369–378. Springer, Heidelberg (1997)
11. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Wonderweb deliverabled18, ontology library (final). laboratory for applied ontology. Technical Report D18 (2003)
12. Mascardi, V., Cord, V., Rosso, P.: A comparison of upper ontologies. In: Proceedings of the Conference on Agenti e industria: Applicazioni tecnologiche degli agenti software, WOA 2007 (2007)

13. Parent, C., Spaccapietra, S., Zimnyi, E.: Spatio-temporal conceptual models: data structures + space + time. In: Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems, Kansas City, Missouri, United States, pp. 26–33. ACM Press, New York (1999)
14. Mark, D.M., Skupin, A., Smith, B.: Features, objects, and other things: Ontological distinctions in the geographic domain. In: Montello, D.R. (ed.) COSIT 2001. LNCS, vol. 2205, pp. 488–502. Springer, Heidelberg (2001)
15. Parnas, D.L.: On the criteria to be used in decomposing systems into modules. Communications of the ACM 15(12), 1053–1058 (1972)
16. Perry, M., Hakimpour, F., Sheth, A.: Analyzing theme, space, and time: an ontology-based approach. In: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, Arlington, Virginia, USA, pp. 147–154. ACM, New York (2006)
17. Yuan, M.: Modeling Semantical, Temporal and Spatial Information in Geographic Information Systems. In: Geographic Information Research: Bridging the Atlantic, pp. 334–347. Taylor & Francis, Abington (1996)
18. Mamei, M., Zambonelli, F.: Programming pervasive and mobile computing applications: The tota approach. ACM Trans. Softw. Eng. Methodol. 18(4), 1–56 (2009)
19. Emmerich, W., Kaveh, N.: Component technologies: Java beans, com, corba, rmi, ejb and the corba component model. In: Proceedings of the 24th International Conference on Software Engineering, ICSE 2002, pp. 691-692 (2002)
20. Lee, C., Nordstedt, D., Helal, S.: Enabling smart spaces with osgi. IEEE Pervasive Computing 2(3), 89–94 (2003)
21. Carriero, N., Gelernter, D.: Linda in context. ACM Commun. 32(4), 444–458 (1989)
22. MacColl, I., Millard, D., Randell, C., Steed, A., Brown, B., Benford, S., Chalmers, M., Conroy, R., Dalton, N., Galani, A., Greenhalgh, C., Michaelides, D., Rodden, T., Taylor, I., Weal, M.: Shared visiting in equator city. In: CVE 2002: Proceedings of the 4th International Conference on Collaborative Virtual Environments, pp. 88–94. ACM, New York (2002)
23. Frey, D., Roman, G.: Context-aware publish subscribe in mobile ad hoc networks. In: Murphy, A.L., Vitek, J. (eds.) COORDINATION 2007. LNCS, vol. 4467, pp. 37–55. Springer, Heidelberg (2007)
24. Cugola, G., Margara, A., Migliavacca, M.: Context-aware publish-subscribe: Model, implementation, and evaluation, pp. 875 –881 (July 2009)
25. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. 5(1), 4–7 (2001)
26. Bernini, D., Micucci, D., Tisato, F.: Space integration services: a platform for space-aware communication. In: Proceedings of the 2010 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. ACM, New York (2010)

Visualising Semantic Coupling among Entities in an OWL Ontology

Juan Garcia, Francisco Garcia, and Roberto Theron

Computer Science Department,
University of Salamanca
{ganajuan,fgarcia,theron}@usal.es

Abstract. An ontology consists of a finite set of entities and relationships among these entities. Relationships in an OWL ontology are given by the object properties, defined as a binary relation between classes in the domain with classes in the range. These relationships define semantic coupling among classes. This paper formally define some metrics to measure semantic coupling among classes in OWL ontologies. Moreover it also describes a visualisation to analyse the proposed metrics using visual analytics techniques. We have implemented a tool to test our metrics with real public ontologies and the results are analysed and discussed.

Keywords: Semantic Coupling, Coupling Metrics, OWL Ontologies, Visual Analytics.

1 Introduction

In general, an ontology describes formally a domain of discourse. Typically, an ontology consists of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain. Relationships typically include hierarchies of classes. A hierarchy specifies a class C to be a subclass of another class C if every ob ject in C is also included in C. Apart from subclass relationships, ontologies may include information such as properties, value restrictions, disjoint statements, specifications of logical relationships between ob jects. We focus on OWL ontologies that distinguish between two main categories of properties: Object properties and Datatype properties. Object properties relate classes in the domain with classes in the range while Datatype properties relate classes in the domain with simple data values in the range. Relationships among classes are given by the Object properties so in order to measure them these properties need to be analysed.

Ontologies are represented in some ontology description languages such as Resource Description Framework (RDF), Web Ontology Language (OWL) and Description Logic (DL). OWL is an extension of RDF and a machine-readable language for sharing and reasoning information on the web. It has been recomended as the standard web ontology language. An OWL ontology representation consists of axioms and facts. Axioms are the semantic knowledge defined

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 90–106, 2010.

by building relationships between classes and properties. Facts represent the individual assertions [8].

Ontologies play an important role to provide shared knowledge models to semantic-driven applications targeted by Semantic Web. Ontology metrics represent an important approach due to they can help to assess and qualify an ontology. From the viewpoint of ontology developers, by assessing quality of ontology, they can automatically recognise areas that might need more work and specify some parts of the ontology that might cause problems. Furthermore metrics are useful in the process of reuse because before using a previously defined ontology would be desirable to evaluate it in order to determine the worthiness of using it. Metrics should always be taken into account to evaluate ontologies both during engineering and application processes.

In the software engineering field, Coupling Between Objects (CBO) has been defined in [1] with two variants. CBO-in that measures the number of classes that depend on the class under consideration and CBO-out that measures the number of classes on which the class under consideration depends. This approach has been taken into account to define our metrics but focused on ontologies.

1.1 Our Contribution

In this paper we formally define the metrics that help to analyse and evaluate coupling between classes in an ontology. These metrics are focused on providing an insight about the importance of the classes according to their relationships and the way they are related. Moreover we also propose a visualisation to analyse coupling using diverse techniques taken from the Visual Analytics field.

This paper is organised as follows: we provided a brief introduction, then we discuss some related work. In the third section we formally define our metrics to measure coupling. In the fourth section we propose a visualisation framework, then we analyse a case study in the fifth section and finally we discuss the conclusions and the future work.

2 Related Work

Diverse ontology metric proposals have been done in the past years; such as [3] that describes some metrics to normalize ontologies and [4][5] that represent a way to rank them. The paper [3] reviews the current state-of-the-art and basically proposes normalization as a pre-process to apply structural metrics. This normalization process consists of five steps: name anonymous classes, name anonymous individuals, classify hierarchically and unify the names, propagate the individuals to the deepest possible classes and finally normalize the object properties. This proposal is focused on content metrics based on OntoMetric framework and basically they have been proposed to improve ontology behaviour or to fix some mistakes. Papers [4][5] propose some metrics to rank ontologies. Basically this proposal consists of a Java Servlet to process as inputs some keywords introduced by the user. Then the framework searches using

Swoogle[1] engine and retrieves all the URI's representing the ontologies related with these keywords. Then the framework searches on its internal database if these ontologies have been previously analysed and retrieves their information. Finally the framework ranks retrieved ontologies.

Orme et al. [6] proposed a set of coupling metrics for ontology-based systems represented in OWL, these metrics are: the number of external classes (NEC), reference to external classes (REC), and referenced includes (RI). This proposal defines a new type of coupling measurement for system development that defines coupling metrics based on ontology data and its structure. The first proposed metric is NEC, representing the number of distinct external classes defined outside the ontology but used to define new classes and properties in the ontology. The external classes can include standard classes defined as ontology language primitives and user-defined classes from other ontologies. The second metric REC is the number of references to external classes in the ontology. As we described above, NEC is a direct measure of the number of classes in the ontology. REC is a direct measure of the number of fanouts (in this case fanouts are different class hierarchies with external roots) within the ontology resulting from external classes. RI is a direct measure of the number of referenced includes in the ontology. Authors have also proposed some cohesion metrics for ontologies [7]. They proposed a set of ontology cohesion metrics to measure the modular relatedness of OWL ontologies. These metrics are Number of Root Classes (NoR), Number of Leaf Classes (NoL) and Average Depth of Inheritance Tree of all Leaf Nodes (ADIT-LN). Authors define NoR metric as the total number of root classes explicitly defined in the ontology. A root class in an ontology means the class has no semantic super class explicitly defined in the ontology. NoL metric is defined as the number of leaf classes explicitly defined in the ontology. A leaf class in an ontology means the class has no semantic subclass explicitly defined in the ontology. Finally ADIT-LN is defined as the sum of depths of all paths divided by the total number of paths. A depth is the total number of nodes starting from the root node to the leaf node in a path. The total number of paths in an ontology is all distinct paths from each root node to each leaf node if there exists an inheritance path from the root node to the leaf node. And root node is the first level in each path.

Yinglong et al. [8] proposed another set of ontology cohesion metrics to measure the modular relatedness of ontologies in the context of dynamic and changing Web. These metrics have been defined taking into account the cohesion principle from Object Oriented Approach adpated to ontologies. Authors concentrate on measuring inconsistencies in ontologies and fully consider the ontological semantics rather than structure. The metrics they propose are Number of Ontology Partitions (NOP), Number of Minimally Inconsistent Subsets (NMIS) and the Average Value of Axiom Inconsistencies (AVAI). This work also describes the algorithms to compute these metrics and validate the metrics by using validation frameworks. These metrics are focused on assesing the quality of ontologies. Authors define NOP metric as the number of semantical partitions of a knowledge

[1] http://swoogle.umbc.edu

base. NMIS is defined as the number of all minimally inconsistent subsets in a knowledge base. This metric is useful to measure the scope of inconsistency impacts of a knowledge base. The third metric AVAI is defined as the ratio of the sum of inconsistency impact values of all axioms and assertions to the cardinality of the knowledge base. Moreover, the article analyses and validates the proposed metrics. Generally speaking, the advantages of these metrics include the possibility of assessing the quality of an consistent ontology.

OntoClean methodology [9], [10] proposes the use of some defined metaproperties. These metaproperties are rigidity, unity, identity and dependency. Authors have borrowed these concepts from their ancient philosophical counterparts. The methodology consists of assigning these metaproperties to the entities in order to provide with a logical and semantic meaning. Applying these metaproperties results on imposing several constraints on the taxonomic structure of an ontology and let to develop a conceptual analysis of the concepts and their validity. Moreover this methodology let to analyse and detect not logically consistent relationships.

YANG et al. [11] proposed metrics from a different point of view. Taking into account the evolution of the ontologies. Authors suggest a metrics suite of complexity, which mainly examine the quantity, ratio and correlativity of concepts and relationships, to evaluate ontologies from the viewpoint of complexity and its evolution. These metrics are divided into two groups: Primitive Metrics and Complexity Metrics. The Primitive metrics include TNOC (Total Numbers of Concepts or Classes), TNOR (Total Number of Relations), TNOP (Total Number of Paths), where a path is defined as a trace that can be taken from a specific particular concept to the most general concept in the ontology. The first Complexity Metric defined is the average relations per concept, that is calculated by dividing TNOR by TNOC. The second metric is the average paths per concept, and is calculated by dividing TNOP by TNOC.

2.1 Tools That Implement Metrics

There are some developed tools that implement diverse metrics. ONTOMETRIC, OntoQA and Protege represent the main available proposals. We consider that OntoQA [12], [13] represents the main proposal about metrics on ontologies. It proposes some Schema Metrics to measure the richness of schema relationships, attributes and schema inheritance. These metrics are focused on evaluating the ontology in general. Another proposed categories are class richness, average population, cohesion, importance of a class, fullness of a class, class inheritance and class relationship richness, connectivity and readability. Class Relationship Richness is defined as the number of relationships that are being used by instances that belong to the class. On the other hand, the Connectivity of a class is defined as the number of instances of other classes that are connected to instances of the selected class. All these metrics are focused on the structure of the ontology.

Currently Protege[2] is the most widely used tool to create or modify an ontology. The metrics are classified into 6 categories. The first one is related with general metrics such as counters for classes, object properties, datatype properties and individuals. The second category is related with class axioms and includes counters for subclass axioms, equivalent classes axioms, disjoint classes axioms, GCI and hidden GCI. The third category includes counters for object properties axioms. These counters are total of sub object properties, equivalent, inverse, disjoint, functional, inverse functional, transitive, symmetric, anti symmetric, reflexive and irreflexive object properties. Furthermore object property domain and range counters are also included. The fourth category is dedicated for datatype properties counters. This category includes total values for sub datatype properties, equivalent, disjoint and functional datatype properties, as well as counters for data properties domain and range. The fifth category is focused on individuals. It defines counters for class assertions, object and datatype property assertions, negative object and negative datatype assertions and same or different individuals axioms. Finally, the last category involves annotation axioms and defines just two metrics, the entity annotation axioms count and the axiom annotation axioms count. All these metrics represent simple counters for the items in the ontology and do not provide any kind of semantic metric.

ONTOMETRIC [14] is a framework proposed to measure the suitability of existing ontologies. This tool was defined to quantify the suitability of ontologies. Authors propose a taxonomy of 160 characteristics, also called multilevel framework of characteristics that provides the outline to be able to choose and to compare existing ontologies. This framework is used as a representation template of the information and starts by defining an analytic hierarchy process. This process involves building a hierarchy tree with the root node being the objective of the problem. The intermediate are the criteria and finally the lowest levels contain the alternatives. Then as the second step, the methodology applies the analytic hierarchy process to decide whether or not to reuse ontologies.

Ontology Metrics [3] is a web-based tool that validates and displays statistics about an OWL ontology, including the expressivity of the language it is written in. This tool calculates the same metrics than Protege. These metrics include counters for classes, properties, individuals, logical axioms, as well as specific counters described above in the Protege section.

2.2 Comparing Metrics

Comparing diverse analysed metrics we consider that OntoQA offers the best set of metrics for analysing the structure and ranking the ontologies according to these metrics. Just Yinglong proposal and OntoClean are clearly focused on the semantics. On the other hand Orme, Yinglong and OntoQA have proposed different metrics to calculate the cohesion. It is important to highlight that they represent three completely different ways to define cohesion. OntoQa defines cohesion as the number of separate connected components of the graph representing the Knowledge Base. In contrast Orme's proposal metrics measure the

[2] Protege Ontology Editor http://protege.stanford.edu/

Table 1. Summarises the analysed metrics and compares some interesting properties

Metric	Semantic / Structure	Ranking	Cohesion	Coupling
Vrandecic	*Structure*	*No*	*No*	*No*
Alani	*Structure*	*Yes*	*No*	*No*
Orme	*Structure*	*No*	*Yes*	*Yes*
Yinglong	*Semantic*	*No*	*Yes*	*No*
OntoClean	*Semantic*	*No*	*No*	*No*
Ontometric	*Structure*	*No*	*No*	*No*
Protege	*Structure*	*No*	*No*	*No*
OntoQA	*Structure*	*Yes*	*Yes*	*No*
OntologyMetrics	*Structure*	*No*	*No*	*No*
Yang	*Structure*	*No*	*No*	*No*

modular relatedness of OWL ontologies while Yinglong focuses on measuring ontologies in the context of dynamic and changing web. The metrics proposed by Orme et al. [6] represent the coupling among entities from diverse ontologies. There is another coupling that has not been taken into account: the coupling among entities in the same ontology that provides useful information about the connection among the classes in the same ontology. Sometimes systems use no more than one ontology then the coupling metrics proposed by Orme become completely useless. Finally, the metrics proposed by Yang et al. [11] are intended to reflect the complexity of an ontology during its lifecycle and evolution.

3 Our Metrics

As we said above relationships between classes in an ontology provide us with an insight of the ontology. The direction of a property can be logically considered going from the domain to the range. This direction provides us important information about the role that classes play in the ontology. For instance a class that belongs to the domain of many properties would represent one of the main subjects of the ontology. This is obvious because it implies that this class is being qualified, evaluated or described. In contrast, a class that belongs to the range of one or many properties would represent a qualifier, characteristic or even more important may be used to infer a certain type, and sometimes these classes represent an enumeration of values. It is important to differentiate between both types of classes in an ontology. Furthermore it is also important to distinguish between a property that has the same class belonging to both domain and range. Our metrics have been defined to be the analogous counterpart from Object- Oriented Systems to ontologies. We define CBE (Coupling Between Entities) for ontologies with two possibilities. CBE-out representing the coupling where the class belongs to the domain of the property and CBE-in representing the coupling where the class belongs to the range of the property. Furthermore we also define SC (Self Coupling) for those properties that have the same class belonging to both domain and range of the property. Figure 1 shows a diagram where CBE is illustrated over class A. The same class may play different roles in an ontology, nevertheless almost always there are some of them that play specific roles being either subjects or qualifiers, depending on the side they belong to.

3.1 Extending the Coupling Metrics

Relationships among entities are also inherited from superclasses to subclasses. In consequence, coupling between entities is also inherited. Our previously defined metrics are focused on direct coupling, it refers to the coupling directly defined in the class without considering the coupling defined in the superclasses. Because of that, we extend the metrics to also consider the inherited coupling from the superclasses. We define iCBE-out as the inherited CBE-out metric, representing the CBE-out metric value from all the superclasses of the selected class. In addition, we also define iCBE-in, iCBE-io and iSC as the CBE-in, CBE-io and SC metrics respectively from all the superclasses of the selected class.

We provide a formal definition of all metrics.

Definitions:

1. $Let\ \Theta\ be\ an\ OWL\ ontology$
2. $Let\ \phi\ be\ the\ set\ of\ properties\ \in\ \Theta$
3. $Let\ C\ be\ the\ set\ of\ classes\ \in\ \Theta$
4. $\exists\, c\ \in\ C$
5. $\exists\, \rho\ \subseteq\ \phi$
6. $\exists\, p\ \in\ \rho\,,\ q\ \in\ \rho\ and\ p\ \neq\ q$
7. $\exists\, d\ \in\ C\ such\ as\ d\ is\ subclass\ of\ c(i)\ \forall\, i\ \in\ [1, n]$
8. $Let\ \alpha(i)\ be\ the\ CBE - out\ metric\ value\ for\ c(i)$
9. $Let\ \beta(i)\ be\ the\ CBE - in\ metric\ value\ for\ c(i)$
10. $Let\ \gamma(i)\ be\ the\ CBE - io\ metric\ value\ for\ c(i)$
11. $Let\ \delta(i)\ be\ the\ SC\ metric\ value\ for\ c(i)$

Metrics:

1. We define CBE-out metric as:

$$if\ c\ \in\ domain(p)\ \forall\, p\ \in\ \rho\ then\ CBE - out\ =\ |\rho|$$

2. We define CBE-in metric as:

$$if\ c\ \in\ range(p)\ \forall\, p\ \in\ \rho\ then\ CBE - in\ =\ |\rho|$$

3. We define CBE-io metric as:

$$if\ \exists\, p\ \in\ \rho\ and\ q\ \in\ \rho\ such\ as\ p\ is\ inverse\ of\ q\ then\ CBE - io\ =\ |\rho|/2$$

4. We define SC (Self-coupling) as:

$$if\ \exists\, c\ \in\ domain(p)\ and\ c\ \in\ range(p)\ \forall\, p\ \in\ \rho\ then\ SC\ =\ |\rho|$$

5. We define iCBE-out metric for class d as:

$$\sum \alpha(i)\ \forall\, i\ \in\ [1, n]$$

6. We define iCBE-in metric for class d as:

$$\sum \beta(i) \; \forall \, i \; \in \; [1, n]$$

7. We define iCBE-io metric for class d as:

$$\sum \gamma(i) \; \forall \, i \; \in \; [1, n]$$

8. We define iSC metric for class d as:

$$\sum \delta(i) \; \forall \, i \; \in \; [1, n]$$

9. Finally we define Coupling Between Entities (CBE) as:

$$\text{CBE} = \text{CBE-out} + \text{CBE-in} + \text{CBE-io} + \text{SC} + \text{iCBE-out} + \text{iCBE-in} + \text{iCBE-io} + \text{iSC}$$

The main advantages of our metrics include: the capacity to detect classes that represent the subject of the ontology and classes that qualify, moreover to be able to discern between both. Furthermore metrics tell us information about how coupled classes in the ontology are, resulting in a low or high coupled ontology. We consider that high coupled ontologies would be desirable because low coupling would imply that classes are not related each other except for ISA relationships.

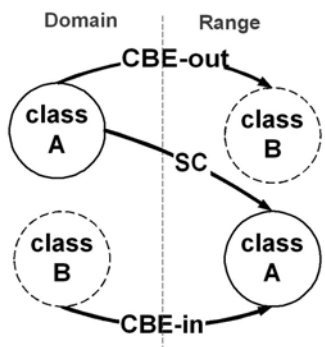


Fig. 1. Shows the CBE-out, CBE-in and SC definitions on class A

Our metrics are semantically defined due to coupling between entities (CBE) involve to define the direction of coupling. This means that semantically speaking CBE-in, CBE-out and SC represent different concepts as explained above in the definition of them. Furthermore changes in CBE values would represent semantic variations not structural. The structure in the ontology is not directly related with our metrics. Opposite to coupling metrics proposed by Orme et al. [6] that relate the external classes, our metrics take into account not just external but internal classes too. External classes refer only to classes defined in other ontologies. Furthermore our proposal is focused on coupling from the point of view of the same coupled classes. We mean that if we take a specific class we represent its own coupling.

4 Our Visualisation Framework

4.1 Framework Architecture, a Visual Analytics Approach

Our framework has been written in Java and uses Jena API to manage OWL ontologies [3]. Jena framework is responsible for loading ontologies. As illustrated on figure 2, our framework queries directly over Jena framework, and builds some data structures to manage the information. This information is processed, managed by the framework and finally visualised. This design was intended to get a framework independent from the API to manage ontologies.

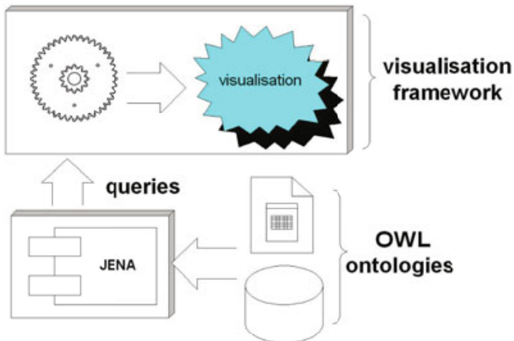


Fig. 2. Shows the framework's architecture

The framework consists on diverse Visual Analytics techniques specialised in covering some aspects of the ontology. These visualisations are interconnected to interact each other. It means that captured events in one visualisation alter the state of other ones. Events include data filtering, or selection of internal items. The term Visual Analytics refers to an analytical reasoning using automated analysis techniques and interactive visualizations to synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data. Our framework includes diverse visualisation techniques, to cover most aspects related with ontologies in a visual manner. The visualisation of metrics defines just one of the aspects to be considered, in our framework.

4.2 Visualising Coupling

Our visualisation is based on a radial layout where selected class is located at the centre, meaning that we are focusing from the point of view of the class. All the properties related with this class are queried and displayed in the radial. Then, for each property the domain and range are analysed in order to detect the classes belonging to each one. According to our definition for CBE, this class is coupled if the CBE value is greater than cero. Furthermore this class is

[3] http://jena.sourceforge.net/

CBE-in or CBE-out coupled according to the place the class belongs to in the property. Coupling is represented in the radial going from left to right as shown in Fig. 3. The radial layout is divided in two sections, having each one a total circumference value of PI. The left side is used to represent CBE-in by showing the properties and the classes in the domain. The selected class belongs to the range of these properties. CBE-out is represented in the right side of the radial in the same way. All the properties are proportionally distributed within an angle equal to PI. Opposite to CBE-in, the selected class belongs to the domain of all these properties, and is located at the left side.

Figure 3 shows a selected class with a CBE-out value of 6 where each property has the selected class in the domain. In consequence each property can have one or more classes in the range. The first property is labeled with hierarchy and illustrates one class in the range and its own subclasses hierarchy. In this case we are representing 6 pink-coloured subclasses inside the class. The second property labeled with cluster represents three classes in the range of the property in a cluster layout like view. The third property labeled with class represents the simplest case where just one class belongs to the range. Finally the fourth property illustrates the way to indicate the attributes of a property; in this case the property is reflexive and symmetric.

Our tool is also intended to highlight the bad definition of concepts in the ontology. For instance a typical and very common error in ontologies occurs when properties are defined without a specific type in the domain or range or neither domain nor range. We consider this a lack of a semantic content due to each property should have clearly defined the types in the domain and range. We highlight this lack by using a symbol that clearly indicates a non-defined type, as illustrated on fig. 3 in the fifth property.

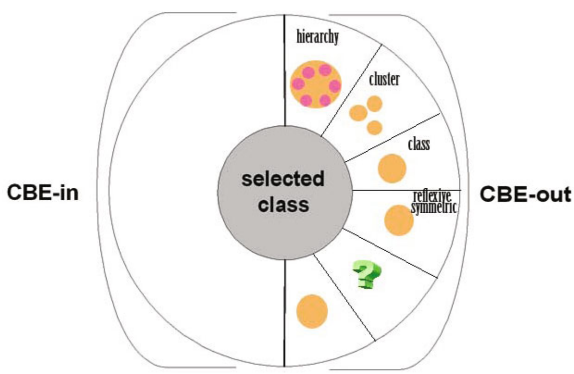


Fig. 3. Shows the general description of our visualisation

4.3 Visualising the Hierarchy and Filtering Data

Hierarchy represents a crucial part of the ontology. We have implemented a treemap visualisation to represent the hierarchy. A treemap is a widely used visual analytics technique proposed by Shneiderman [18]. It is a space-filling visualisation technique that represents large hierarchical data sets via a two-dimensional rectangular map, providing a compact visual representation of complex data spaces through both area and color. The area is related with the internal hierarchy it means the total of subclasses, while the color is related with properties of classes such as transitive, symetric, inverse, functional, etc.

Metrics are represented using another technique called table lens [19]. Table Lens technique has been motivated by the particular nature of tables. This technique is used to represent all the measured metric values and it is basically used to filter data. Criteria to filter data is based on metrics such as cbe, sc or the number of properties. Filtering is very important because when we are analysing coupling we are just interested in coupled classes, and the tool provides a way to filter these classes. Left side on figures 7(a) and 8 show a table lens where a filter on classes has been applied. The classes with a cbe value greater or equal than one have been filtered and descending ordered. This let us to recognise the coupled classes and specifically the most coupled one which in both cases is the class Person.

5 Case Study

We started our case study by searching some public available ontologies to analyse. The easiest way to search for ontologies is by using the Swoogle Semantic Web Search Engine [4]. We searched for ontologies related with the keyword "*person*", we consider it is the most basic word to search for. After a selection process and discarding those ones bad-defined, we selected five OWL ontologies to work with. Each one are described below.

5.1 SWETO

The first analysed ontology is Semantic Web Technology Evaluation Ontology (SWETO) version 1.4 [5] [16]. This general-purpose ontology was developed in 2004 by the Computer Science Department at the University of Georgia. This SWETO version consists of 114 classes and 13 object properties. Table 1 shows the result of applying our metrics to this ontology and all the classes that have at least 1 coupling value.

The simple analysis of SWETO coupling metrics has shown interesting results. Even having no background information about the ontology, we can clearly deduce the most coupled class also represents the most important one, in this case 'Person' depicted on figure 4(b). This is an obvious result because if this class contains most of the object properties then it represents the main ontology's subject. Furthermore this result can be supported by the fact that all the relationships are in the CBE-out metric which means that this class belongs to the domain in all the object properties. This fact implies these object properties were defined having as main purpose to satisfy some needs for this specific class. On

[4] http://swoogle.umbc.edu/

[5] http://knoesis.wright.edu/library/ontologies/sweto/

Table 2. Summarises the proposed coupling metrics in the ontology SWETO

Class	Object properties	CBE-out	CBE-in	CBE-io	SC	CBE
$Person$	5	5	0	0	0	5
$Organization$	1	1	3	0	0	4
$Publication$	1	1	2	0	1	4
$Place$	0	0	3	0	0	3
$AcademicDepartment$	0	0	2	0	0	2
$Event$	1	1	1	0	0	2
$Researcher$	1	1	0	0	0	1
$University$	1	1	0	0	0	1
$Proffesor$	1	1	0	0	0	1
$ScientificPublication$	1	1	0	0	0	1
$Thing$	1	1	0	0	0	1
$Country$	0	0	1	0	0	1
$Classification$	0	0	1	0	0	1

the other hand, classes that only have CBE-in values such as Place, Academic Department, Country or Classification belong to the range in the properties. It means that individuals of these classes represent all the possible values the properties can take in the range. Moreover these individuals represent qualifiers for other entities. Analysing CBE-io metric we realise that there are no values greater than 0. It means there are no inverse functions declared in the ontology. Moreover there is only one class with self coupling value (Publication); meaning that exists one property which has the same class in the domain and range. This analysis let us to have an insight into the behaviour of the ontology even without previous knowledge background of it. The main purpose is to relate persons with places, organizations or academic departments. Furthermore more specialized classes such as Professor or Researcher are related with Publication class depicted on figure 4(c). We can deduce from this ontology that it shows low coupling between classes. Just 13 out of 114 classes are related, which represents less than 10 percent of the total.

5.2 C2IEDM

The second analysed ontology is C2IEDMTrackCoreOnt.owl [17] downloaded from VIStology corporation website [6] designed to be used in military operational environments. This ontology is an implementation of the Command and Control Information Exchange Data Model (C2IEDM) developed by Multilateral Interoperability Programme (MIP). This is a small ontology, being the most coupled class OBJECT-ITEM-LOCATION which is depicted on figure 5(b). The property object-item-location-bearing-angle has no defined type class in the range as can be seen on the same figure 5(b). Figure 5(c) depicts the coupling for the class REPORTING-DATA where it can be seen that the functional property 'reporting-data-id' has more than one class in the domain, specifically three.

[6] http://www.vistology.com/index.html

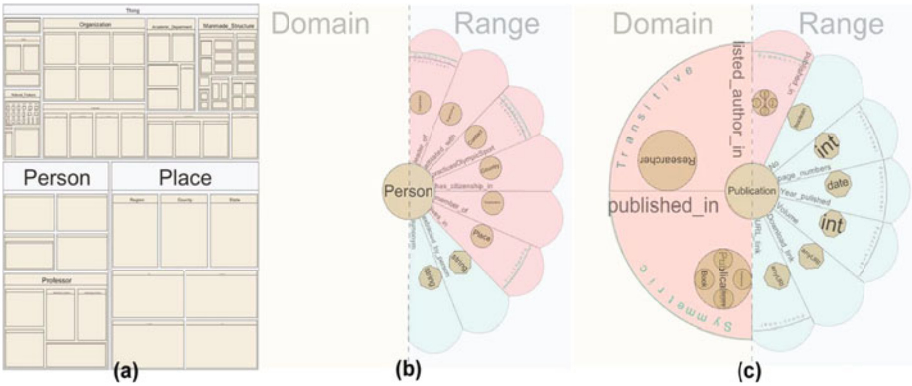


Fig. 4. Fig. 4(a) shows the hierarchy of the ontology represented in a treemap visualisation. Fig. 4(b) shows coupling for class Person and Fig. 4(c) shows coupling for Publication class.

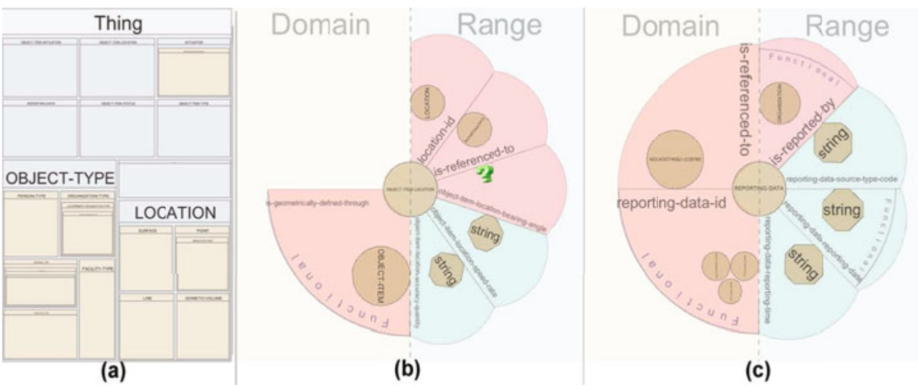


Fig. 5. Fig. 5(a) shows the hierarchy of the ontology represented in a treemap visualisation. Fig. 5(b) shows the most coupled class OBJECT-ITEM-LOCATION. Fig. 5(c) shows the coupling for class REPORTING-DATA.

They are represented as a cluster in the domain of the property. This ontology has just 25 classes and 8 of them with cbe value equal or greater than one, representing 32 percent of coupled classes.

5.3 JC3IEDM3

The ontology JC3IEDM3.1a.owl [17] developed by VIStology corporation is an implementation of the Joint Command, Control and Communication Information Exchange Data Model (JC3IEDM). JC3IEDM is being developed by the NATO-sponsored Multilateral Interoperability Programme (MIP) for the purpose of facilitating the exchange of command, control and communication

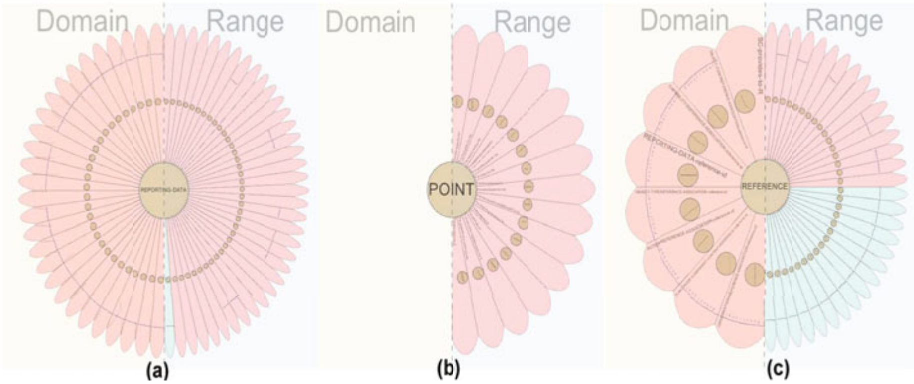


Fig. 6. Fig. 6(a) shows the class REPORTING-DATA which is the most coupled class in the ontology. Fig. 6(b) shows the class Point which is a typical class with CBE-out coupling. Fig. 6(c) shows the class REFERENCE one of the most coupled classes.

information among coalition forces. It represents a modified updated version from C2IEDM. This is a very extensive ontology containing over 7900 elements (OWL classes, properties and their instances).

REFERENCE is shown in the figure 6c. It can be seen that most of the properties are functional, which means that they can have only one unique value for each instance. This OWL ontology was automatically generated from the ERwin data model containing the C3 Information Exchange Data Model. As consequence, this is a well-defined ontology, all properties have been defined with one class in the domain and range, being the most coupled class REPORTING-DATA shown in the figure 6c. This class has a CBE total value of 76 according to 36 for CBE-out, 28 for CBE-in and 12 for CBE-io and 1 datatype. It is a very high coupled ontology in contrast to C2IEDM and SWETO. This ontology is very crowded with 1488 classes and also very coupled. There is no a minimal percentage of coupling for a class, nevertheless high coupling is always desirable in contrast to low coupling.

5.4 Portal

The ontology portal.owl [7] taken from ontoware server is a general purpose related with some aspects of technology and academics. We emphasise on figure 7(a) the data filtering according to cbe metric value. Analysis is focused on coupling among classes, so the first step is to filter classes to only keep the coupled ones, this is done using the table lens visualisation. Then all the filtered classes are descending ordered according to cbe value to stablish coupling levels for each one. The result is illustrated on figure 7(a) where it can be depicted just the filtered classes and coupling for class Person. In contrast, figure 7(b) depicts the coupling for the class Generic-Agent which as can be clearly seen has cbe-in

[7] http://ontoware.org/swrc/portal.owl

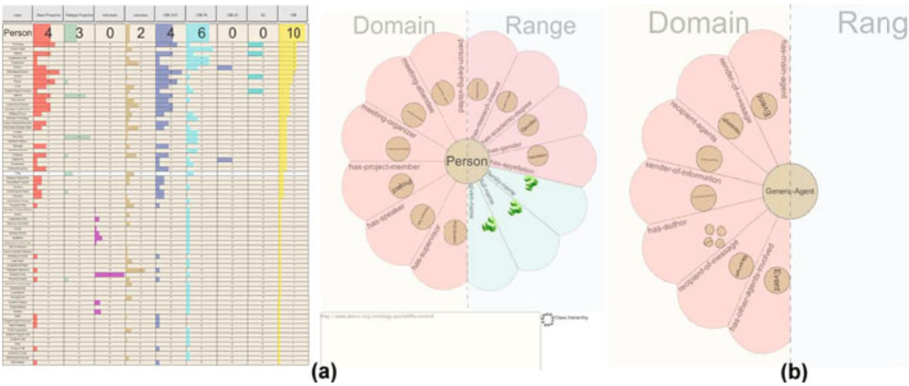
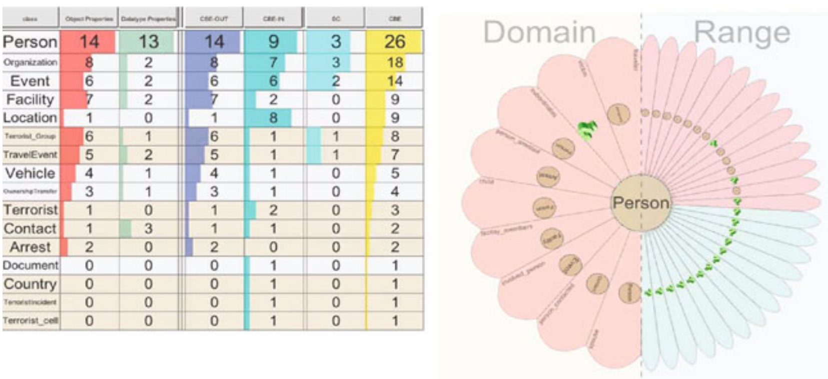


Fig. 7. Fig. 7(a) shows the table lens with the coupled classes filtered, and selected the most coupled class Person. Fig. 7(b) shows the class Generic-Agent which has a value for cbe-in metric of 7.



class	Object Properties	Datatype Properties	CBE-OUT	CBE-IN	SC	CBE
Person	14	13	14	9	3	26
Organization	8	2	8	7	3	18
Event	6	2	6	6	2	14
Facility	7	2	7	2	0	9
Location	1	0	1	8	0	9
Terrorist_Group	6	1	6	1	1	8
TravelEvent	5	2	5	1	1	7
Vehicle	4	1	4	1	0	5
OwnershipTransfer	3	1	3	1	0	4
Terrorist	1	0	1	2	0	3
Contact	1	3	1	1	0	2
Arrest	2	0	2	0	0	2
Document	0	0	0	1	0	1
Country	0	0	0	1	0	1
TerroristIncident	0	0	0	1	0	1
Terrorist_cell	0	0	0	1	0	1

Fig. 8. Fig. 8 shows coupled classes in the ontology filtered by the metric cbe in a table lens, and the coupling of the class Person, that as can be seen is the most coupled class in the ontology

coupling value of 7. This class is an example of a class with only cbe-in coupling. This ontology has 168 classes and coupling percentage of almost 55 percent which is consider acceptable.

5.5 Terrorism

The ontology terrorOnt.owl [8] published by the mindswap group was also used to analyse coupling. This ontology is very coupled due to 16 out of 30 classes have cbe values equal or greater than one, representing 53 percent of coupled classes. In contrast there are many properties with no definitions as can be seen in figure 8 remarked with an interrogation sign. In fact all the datatype properties

[8] http://www.mindswap.org/2004/terrorOnt.owl

have no defined a specific type. We consider this fact as a lack of information and a bad defined ontology. A weakly typed ontology is easier to become inconsistent with the data. Data inconsistencies are not desirable due to they could affect either syntactically or semantically to the ontology.

6 Conclusions and Future Work

We have formally proposed some metrics to measure coupling among classes in OWL ontologies. These metrics are focused on semantic coupling that as we have shown it is important to be considered. Our proposals are based on the widely used CBO (Coupling Between Objects) metrics taken from the Object Oriented Approach, but modified to be applied to OWL ontologies. To illustrate our metrics and our visualisation, we analysed diverse ontologies based on a search related with the subject Person. As it would be expected the class Person in all these ontologies is either the most or one of the most coupled classes. On the other hand, we developed a visualisation that clearly represents our metrics and highlights some aspects such as lack of definitions. We have illustrated and described our visualisation with the analysed ontologies to support the analysis process of coupling. The future work includes more proposal of metrics and new visualisations of these metrics, as well as the integration of this visualisation in a whole visualisation framework for ontoologies.

References

1. Chidamber, S., Kemerer, C.: A Metrics suite for object Oriented Design. IEEE Transactions on Software Engineering 20(6), 476–493 (1994)
2. Harrison, R., Counsell, S., Nithi, R.: Coupling Metrics for Object-Oriented Design. In: Proceedings of the 5th International Symposium on Software Metrics. IEEE Computer Society, Los Alamitos (1998)
3. Vrandecic, D., Sure, Y.: How to Design Better Ontology Metrics. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 311–325. Springer, Heidelberg (2007)
4. Alani, H., Brewster, C., Shadbolt, N.: Ranking Ontologies with AKTiveRank. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 1–15. Springer, Heidelberg (2006)
5. Alani, H., Brewster, C.: Metrics for Ranking Ontologies. In: 4th Int. EON Workshop, 15th Int. World Wide Web Conference (2006)
6. Orme, A., Yao, H., Etzkorn, L.: Coupling Metrics for Ontology-Based Systems. IEEE Software, 102–108 (2006)
7. Yao, H., Orme, A., Etzkorn, L.: Cohesion Metrics for Ontology Design and Application. Journal of Computer Science 1(1), 107–113 (2005)
8. Y. Ma, B. Jin, Y. Feng, Semantic oriented ontology cohesion metrics for ontology-based systems. The Journal of Systems and Software (2009)
9. Guarino, N., Welty, C.: An Overview of OntoClean. In: The Handbook on Ontologies, pp. 151–172. Springer, Berlin (2004)

10. Guarino, N., Welty, C.: Evaluating Ontological Decisions with OntoClean. Communications of the ACM, 61–65 (2002)
11. Yang, Z., Zhang, D., Ye, C.: Evaluation Metrics for Ontology Complexity and Evolution Analysis. In: IEEE International Conference on e-Business Engineering (ICEBE 2006) (2006)
12. Tartir, S., Arpinar, B., Moore, M., Sheth, A., Aleman-meza, B.: OntoQA: Metric-based ontology quality analysis. In: CiteSeerX - Scientific Literature Digital Library and Search Engine (2005)
13. Tartir, S., Arpinar, B.: Ontology Evaluation and Ranking using OntoQA. In: Proceedings of the International Conference on Semantic Computing (2007)
14. Lozano-Tello, A., Gomez-Perez, A.: ONTOMETRIC: A Method to Choose the Appropiate Ontology. Journal of Database Management (2004)
15. Antoniou, G., van Harmelen, F.: A Semantic Web Primer, 2nd edn. The MIT Press, Cambridge (2008)
16. Aleman-meza, B., Halaschek, C., Sheth, A., Arpinar, B., Sannapareddy, G.: SWETO: Large-Scale Semantic Web Testbed. In: 16th International Conference on Software Engineering and Knowledge Engineering (2004)
17. Carlton, B., Farmer, G., Scrudder, R.: A C2IEDM Based Approach to C4I and Simulation Initialization and Synchronization. In: The Interservice/Industry Training, Simulation and Education Conference, I/ITSEC (2003)
18. Shneiderman, B.: Tree visualization with treemaps: a 2-d space-filling approach. ACM Transactions on Graphics 11(1), 92–99 (1992)
19. Rao, R., Card, S.: The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Boston, MA (1994)

Query Expansion for the Legal Domain: A Case Study from the JUMAS Project

Fabio Sartori and Matteo Palmonari

Department of Computer Science, Systems and Communication (DISCo)
University of Milan - Bicocca
viale Sarca, 336
20126 - Milan, Italy
Tel.: +39 02 64487913, Fax: +39 02 64487839
{sartori,palmonari}@disco.unimib.it

Abstract. This paper presents an ongoing research on the comparison of ontological query expansion methods. Query Expansion is a technique that aims to enhance the results of a search by adding terms to the search query; today, it is a very important research topic in the semantic web and information retrieval areas. Although many efforts have been made from the theoretical point of view to implement effective and general methods for expanding queries, based on both statistical and ontological approaches, the practical applicability of them is nowadays restricted to few and very specific domains. The aim of this paper is the definition of a platform for the implementation of a subset of such methods, in order to make comparisons among them and try to define how and when use ontological QE. This work is part of JUMAS, a research project funded by European Community where query expansion is used to support the retrieval of significant information from audio–video transcriptions in the legal domain.

Keywords: Ontological Query Expansion.

1 Introduction

The legal domain has been of interest to Artificial Intelligence (AI) for many years. A number of studies in AI and Law have been produced, which are based on ontological assumptions drawn from legal theory.

For instance, Allen and Saxon [1] have developed a language for legal relations representations in which they have translated the Hofheldian approach to law[1] into about forty relations. Mamfelt [2] proposed a framework where Hart's primary and secondary rules[2] were divided into meta–levels of logic programming. Valente and Breuker [3] have formalized legal knowledge in the ON–LINE system mainly taking into account concepts expressed by Kelsen and Hart.

[1] See e.g. http://www.yale.edu/lawweb/jbalkin/articles/hohfeldiansemiotics.pdf.
[2] See e.g. http://legaltheorylexicon.blogspot.com/2004/06/legal-theory-lexicon-039-primary-and.html.

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 107–122, 2010.

In literature, it is possible to identify many legal ontologies that have been used in practice, like LLD (*Language for Legal Discourse*) [4], *NORMA*, [5] [6], FOLaw (*Functional Ontology of Law*), FBO (*Frame-Based Ontology of Law*) [7] and LRI (*Core Legal Ontology*) [8]. Moreover, legal ontologies have been widely used in the recent past to support the the formalization of legal knowledge and create useful tools. For example, they have found application in many projects founded by the European Commission (EC), like CLIME [9], e–Power [10], e–COURT [11]. These projects are mainly devoted to disambiguate legal jargon except the CLIME one, in which also the legal problem solving strategy is partially implemented.

The purpose of this work is the description of a part of the JUMAS system, a research project funded by EC under the 7th Frame Programme[3], where ontologies are exploited to build a complete knowledge management system for legal operators (e.g. judges, prosecutors, witnesses).

A Query Expansion (QE) Platform based on a thesaurus of legal terms has been designed and implemented to allow operators to browse process transcriptions and audio–video data acquired from courtrooms. QE is a technique that aims to enhance the results of a search by adding terms to the search query. It is useful because it can deal with the inherent ambiguity of natural language, indeed every language has problems of synonymy (more terms for a concept) and polysemy (more concepts for a term) that cause a decay of performance: the search engine doesn't return relevant documents that are present in the domain (low recall) and returns documents that are not relevant (low precision). Query expansion aims to increase the number of relevant documents retrieved.

There are two main approaches to accomplish this task [12]: probabilistic query expansion and ontological query expansion. The first is based on statistic data that indicates the frequency of terms in a collection of supposed relevant documents, or the frequency of co-occurence of terms. The second is based on knowledge models, in particular ontologies, in which the method search for the terms that have to be added to the query.

Although probabilistic query expansion is the dominant approach, it has shown limits in different fields and tasks: Carpineto et al. [13] highlight how this approach weights the terms on the basis of their relevance within the text rather than the real benefits for the user; moreover, it has been pointed out [14][15] that probabilistic methods are very influenced by the corpus of documents and their relevance to the query.

On the contrary, ontological approaches are less developed and studied, but they virtually have a lot of undiscovered possibilities to semantically improve queries: being *corpus-independent*, they are more precise than probabilistic methods in the text disambiguation within a given domain. Moreover, they are particularly suitable to treat short queries. Anyway [16] [17], ontological methods have some important drawbacks: ontologies are typically difficult to create and maintain in order to guarantee the necessary level of precision to avoid the decrease of performance. The so called *query drift* (i.e. the choice of an expansion direction that is

[3] Grant agreement nr. FP7-214306.

out of user scope) phenomenon is more probable with ontological approaches than probabilistic ones.

Our work has been initially focused on the comparison of ontological query expansion methods in an analytic way. Then, we implemented a generic testing platform that allows to easily create, test and compare methods with different ontologies and on different domains.

The platform has been finally adopted in the context of the JUMAS project, where a thesaurus of legal terms has been used as the source ontology for query expansion. The results of QE in JUMAS are exploited from other modules of the system, in particular the audio and video summarization ones.

The paper is organized as follows: Section 2 briefly introduces the context and the aims of the JUMAS project; Section 3 concerns a brief review of literature about query expansion and ontological methods for making it as well as the design and implementation issues we took into account for developing a QE platform for JUMAS. An example of platform application to a thesaurus developed for the JUMAS project is briefly introduced in Section 5, where some preliminary considerations about the effectiveness of our platform are also presented. Finally, conclusions and future work are briefly pointed out in Section 6.

2 The JUMAS Project

Recent technological advances in multimedia production, storage and distribution have moved to digitize the traditional analogical archives creating large digital libraries. This will increase the need of tools for content analysis, information extraction and retrieval of multimedia objects in their native form in order to make easier both the access and the management of multimedia archives. In the last years huge advances have been done in the context of text document processing and several researchers have been working for finding more efficient algorithms for retrieval and information extraction from textual sources.

On the other hand starting from the 80s Audio&Video have been attracting the attention of an increasing number of researchers working on segmentation, clustering and topic identification in Multimedia Document. Using an automatic speech recognition system to transcribe captured speech would leverage its potential value as an information source by making it available for use in the same way as written documents. Capturing the information that people exchange in meetings, during telephone conversations, and in public settings like trial processes and council meetings, could provide access to knowledge whose preservation is of crucial importance.

The experience with applying several documents-based tasks to multimedia document indicates that there is potential for success in this endeavor. In particular, in order to apply traditional retrieval methods to spoken document, the current solution is to produce a manual verbatim transcription performing then the effective consultation process on the generated text. In contrast, the system proposed in this project provides an automatic fusion and integration of audio, video and text sources where high-quality automatic transcription is provided by continuous semantic feedback between the media and text analysis components.

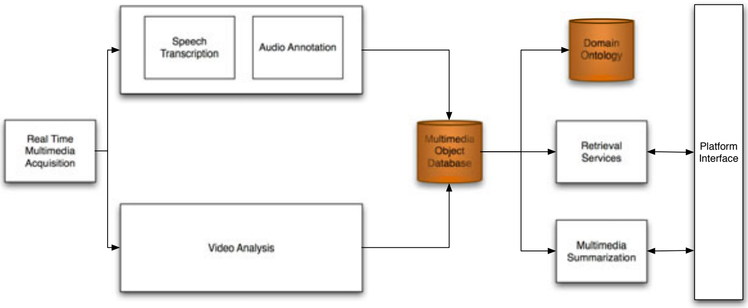


Fig. 1. The Architecture of the JUMAS System

JUMAS contribution (see Figure 1) is focused on supporting the workflow management and the semantic interpretation of recorded audio-video streams by developing new models and techniques for representing and automatically extracting the embedded semantics derived from multiple data sources. JUMAS will be used for collecting, sharing, annotating and retrieving information from multimedia data streams typical of video-recorded administrative and public debates.

The most important goal of the JUMAS system is to collect, enrich and share multimedia document annotated with embedded semantic minimizing manual transcription activity. These complex activities will be performed in order to assist the user with information extraction, text mining and retrieval tools to improve information production, exploitation and interpretation. Following these objectives the key functionalities of the proposed system are:

- Automatic Transcription: Today the bottleneck of the consultation process is the production of text associated to the stream audio/video, carried out by manual transcriptions. Automatic processes of information transcription are necessary and helpful for the end-users during the consultation phases.
- Semantic Annotation: The exploitation of knowledge is spread across information in different media. Using a fusion of information derived from audio transcription and video analysis can compensate eventually corrupted sources, define more fine grained annotation concepts, allow user-friendly browsing, etc. Video annotation (assigning semantic values to scene events) is based on the segmented objects and events, considering a given scene geometry and the possible rules of the legal process. Annotation is a high level process where visual information of the previous feature extraction steps are connected to the textual (from speech) and scene (geometry, position, motion) information. The annotation also needs a learning environment to register the situation of the detected objects.
- Retrieval Services: Currently the process of audio/video segments retrieval needs the manual consultation of the entire tracks on which it has been recorded part of a specific spoken document. Today the identification of a particular position on multimedia stream, with the aim to look at/to listen specific declarations, participations, testimonies, etc..., is possible only

remembering the temporal instant in which events happened. For this reason, the proposed system based on an automatic transcription and semantic annotation components, will allow the construction of a flexible retrieval environment not only based on simple textual queries, but on wide and complex concepts. In order to define an integrated platform for cross-modal access to audio, video recordings and their automatic transcript, models capable to perform semantic multimedia indexing and retrieval will be developed.

The subject of this paper is the description of the domain ontology part of the JUMAS architecture. As shown in Figure 1 this a module that works on the transcription produced by audio and video summarization to support the retrieval methods implemented to guide operators in finding useful information and knowledge used in the past.

3 Query Expansion: State of the Art and Our Approach

In the analysis of literature we first explored the various definition of query expansion[18][19][20][21] and we tried to merge them to obtain a possible common definition that contains the main characteristic of each vision. Indeed, there are several different definitions because the problem of query expansion has been approached by different ways and different points of view. However the goals and basic mechanisms of expansion are common: to improve the effectiveness of a search, that is finding more relevant documents and less irrelevant documents, by adding new terms to the original search query.

We can consider query expansion as a system that accepts in input a set of words Q returning a new set Q' as the union between Q the set of expanded words E:

$$Q \implies \boxed{\text{QEXP}} \implies Q' = Q \cup E$$

Words belonging to E are functions of a subset of Q, i.e. each element of E has been obtained starting from an element of Q.

$$E = \{ \ \omega_e \ : \ \omega_e \ \in f(\ W_e \subseteq Q \ ), W_e \subseteq Q \ \}$$

More or less, all the existing approaches perform query expansion on the basis of one and only one word of Q:

$$E = \{ \ \omega_e \ : \ \omega_e \ \in g(\ \omega_i \in Q \ ), w_i \in Q \ \}$$

The general model of QE can be further developed according to the following constraints:

– the Q and Q' set can be *ordered* or *not ordered*. In the first case, the order according to which the different words are arranged must also be considered;

– the Q' set can be a collection of pairs (w, p), where w is the word to expand and p is a *weight* related to it. This kind of output is useful when the result of query expansion must be returned to a research engine accepting weighted terms in input. The output would become a set Q'' as follows:

$$Q'' = \{\ (\omega, p)\ :\ \omega \in Q',\ p \in \{0, 1\}\ \}$$

– the output can be the result of a rewriting of Q' according a given rule r. For example, if the research engine supports operators like $\wedge(AND)$ e $\vee(OR)$, we could obtain a rewriting similar to the following one:

$$(\omega_1 \vee \omega_{1,1} \vee \omega_{1,2} \vee \ldots) \wedge (\omega_2 \vee \omega_{2,1} \vee \omega_{2,2} \vee \ldots) \wedge \ldots$$

where $\omega_{i,e}$ are all the expanded terms of the word ω_i.

The first step in the design and implementation of a QE platform for the JUMAS project was a literature review to find the most suitable ones for our scope. As briefly introduced above, QE can be made exploiting both statistical and ontological methods.

Ontological methods are mainly based on the adoption of *thesauri* as the source ontology, like e.g. in the approach (i.e. *Voorhees* in the following) defined by Voorhees [14], where synonyms and any other kinds of term are added to the query linking them directly to the original term. The ONKI method [12] allows the user to select interactively by means of a thesaurus the concepts to search for. Then, it expands automatically the keywords exploiting the same thesaurus. The method is also interesting because it is *multi–ontology*: the *Finnish Collaborative Holistic Ontology (KOKO)* is used for expanding the concepts and the *Finnish Spatio-temporal Ontology (SAPO)* for expanding the places; Alani et al. [22][23] developed an approach (i.e *ATJ* in the following) which expands a query on every direction within the ontology, assigning different weights according to the nature of the followed relationship; Navigli&Velardi [16] defined two methods for QE: in the first one, ontologies are used to extract the concept descriptions. The similarity between two concepts is then evaluated on the basis of the similarity value between their descriptions. The second method is based on the creation of semantic nets of candidate concepts starting from a given ontology, to chose in a second step the terms according to how much such nets overlap. Finally, in [24] an approach to QE (i.e. *HO* in the following) is described that considers the sequences of links through which the terms are bounded to the query terms, taking especially care of terms whose links satisfy given patterns.

Moreover, there exist hybrid systems that exploit ontologies during one or more expansion phases: the method by Andreou [25] has a first step in which an expansion based on probabilistic methods is done, and a second one during which an ontology is used to accomplish a revision of terms, getting better results; the approach proposed by Xhu et al. [18] exploits the *Latent Semantic Indexing* technique, comparing *local* and *global* expansion methods. Here, ontologies allow weighting the terms; Calegari and Pasi [26] exploited a local fuzzy ontology to keep trace of past searches made by the user, with the aim to expand searches with the terms that are most frequently inserted in the same query.

Among these methods, it has been decided to implement three of them: **Voorhees**, **ATJ** and **HO**. In addition, the platform includes **Ancestor**, an approach that calculates the distance between the meanings of two concepts in terms of their distance from a common parent in a tree–like structure.

4 Towards a Platform for QE in JUMAS: Implementation

In the second step, we implemented a generic platform for testing query expansion, and also the previously selected methods. We first defined requirements and motivations for the platform, then we discussed all the implementation choices made during the process. One of the most important choice concerned the decision of which type of ontology to use: we decided to use a *thesaurus*, a restriction of ontologies expressly defined for lexical networks of terms. Next, we'll describe the architecture of the platform, that has been implemented in *Java*. In the implementation process we paid attention to fulfill the requirements and provide the platform with the necessary characteristics: *generality*, *expandability*, *modularity* and *uniformity*. The platform has been exploited to codify the four methods introduced above.

4.1 Type of Ontologies: Thesauri

Thesauri are structured vocabularies where relationships among terms are specified. They can be considered as a subclass of ontologies, since they represent only a part of the knowledge involved and their power of expression is limited [27]. Anyway, their main features are compatible with most of the ontological methods for QE available at the moment.

There exist two *ISO* standards for the definition of thesauri, the *ISO2788* for single–language thesauri and *ISO5968* for multi–language thesauri, which define the possible relationships among the terms of a thesaurus:

USE / UF (Use For) are two relationships for solving synonymy problems; the *USE* relationships specifies that the term considered is linked to another one that would be preferable (i.e. its synonym), while the *UF* relationship is the inverse of USE. These relationships generate networks are *synonyms clusters* which are be referred to as *synsets* within *WordNet*[4], probably the most known English thesaurus;

BT (Broader Term) is a relationship used for specifying more generic terms. Its inverse is the *NT (Narrower Term)*. relationship. BT and NT allows the definition of hierarchical structures among the terms and the synonym clusters;

RT (Related Term): is a relationship which describes a generic association between two terms. This association is neither USE / UF nor BT / NT.

Moreover, the two standards define the following property:

SN (Scope Note): is used to bind a term and its textual description, with the aim to make not ambiguous the usage of such term.

[4] http://wordnet.princeton.edu/

The relationships above are sufficient to create a network of terms which can be very useful in the QE process: for this reason, many approaches are based on WordNet. Moreover, thesauri are simpler to implement and maintain than ontologies and they have no significant drawbacks with respect to the QE approaches we have chosen to implement. For this reasons, it has been decided to design the platform for QE on the basis of thesauri, paying attention to an opportune strategy for adapting it to ontologies adoption in the future.

4.2 Definition of the Ontology Representation Language: SKOS

The initial idea to exploit *OWL (Ontology Web language* [5] for the ontology representation has been modified with the restriction to thesauri. For this reason, it has been chosen to withdraw to *SKOS* (Simple Knowledge Organization System)[6] a formal language that is going to become the standard de facto for thesauri representation.

As the OWL language, SKOS is based on *RDF*: this is very important from the adaptability of our platform to more general ontologies in the future. As *RDF*, *SKOS* is a language for the definition of graphs (in this case graphs of concepts) by means of 3–ples $\langle subject, predicate, object \rangle$. The *predicate* represents the relationship between the *subject* and the *object* .

The elements of the graph are the *concepts* expressed by the terms, which belong to a specific class named *skos:Concept*. The whole graph is described by the *skos:conceptScheme* class.

The kinds of relationships (i.e. the *predicates*) and properties fulfill very well the standards previously described; in fact:

– *skos:prefLabel* and *skos:altLabel* allow modeling the synonymy: every concept is described by a *prefLabel*, which is the preferred term to express that concept and a sequence of alternative terms named *altLabel*, which can be considered synonyms. In this way, an opportune *synset*is defined;
– *skos:broader* and *skos:narrower* represent the *BT (Broader Term)* and *NT (Narrower Term)* relationships;
– *skos:related* represents the *RT (Related Term)* relationship;
– *skos:scopeNote* is a representation of the Scope Note property.

Moreover, *SKOS* introduces other relationships and properties: for example, the*skos:definition* property is exploited to give a precise definition of a term.

4.3 Platform Architecture

Figure 2 shows a sketch of the platform architecture, which is divided into three main blocks responsible for the different functionalities:

– the **Query Expansion Testing Interface** implements generic functions to interact with the user;

[5] `http://www.w3.org`
[6] `http://www.w3.org/2004/02/skos/`

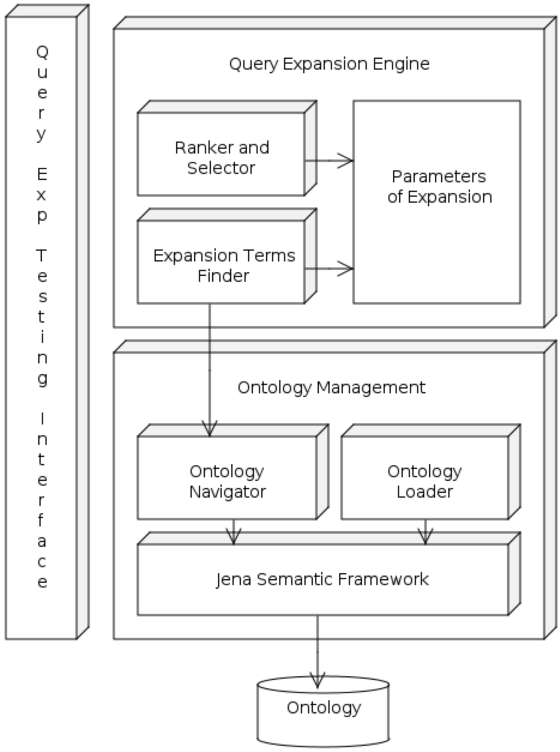


Fig. 2. The Architecture of the QE Platform

- the **Query Expansion Engine** implements the engine for the expansion, which is the specific query expansion method chosen among the four approaches introduced above. This component is made of an *Expansion Terms Finder* (ETF) that looks for the closest terms to the subject of QE and a *Ranker and Selector* that focuses on the following expansion phases, weighting the terms returned by ETF and adding the most significant ones to the QE output set;
- the **Ontology Management** allow managing the ontologies by means of Jena, a semantic framework that provides the system with many tools to treat ontologies and RDF language.

  Moreover, also an *Ontology Loader* and an *Ontology Navigator* are supplied: the first component allows the system to load ontologies from files or other sources, the second one allows searching within the ontology the terms to add to the candidate set by means of SPARQL queries.

The platform has been designed to be an effective and flexible tool to test the different methods, to modify their parameters and to combine sets of methods for deriving new approaches to QE.

5 A Case Study: Expanding Legal Terms in the JUMAS Context

In this section we present the results of the collaboration between the Research Centre on Complex Systems and Artificial Intelligence (CSAI) of the University of Milan-Bicocca and Project Automation in the context of the JUMAS project (Judicial Management by Digital Libraries Semantics), co-funded by the European Community (grant agreement number FP7-214306) within the 7th Frame Programme.

The aim of the collaboration was the development of a legal ontology for supporting the retrieval methods developed by the JUMAS Consortium to extract useful information from audio-video transcriptions of legal acts and processes. In this sense, it was decided to design and implement a software platform for the ontological query expansion (QE) of Italian and Polish legal terms, the target judicial systems taken into account by the project. The section illustrates different steps through which this platform has been developed.

The query expansion module (see Figure 3) is a web service, realized with the JAX-WS 2 (Java API for XML Web Services) technology, which is part of the Java EE 5 platform. The chosen semantic framework is Jena , an open source Java framework for building Semantic Web applications grown out of work with the HP Semantic Web. It provides a programmatic environment for OWL ontologies and includes a rule-based inference engine. Jena provides persistent storage of ontologies in relational databases, through RDBMS back–end adapters. It is possible at runtime to download the ontology, edit the ontology with and external tool (e.g. Protègè) and reload it on the repository. The query expansion engine is the main component of the query module. It is composed of two sub-modules: the Ontology Explorer and the Ranker. The aim of the Ontology Explorer is, starting from a given term, explore the ontology graph and search the terms the are related. The output of the Ontology Explorer is a graph, that is a sub-graph of the ontology. The size of the graph depends on the searched term, the max size of the sub-graph is configurable. The ranker takes as an argument the sub-graph and produces a weighted list of related terms.

In order to expand the query, it is necessary to find in the thesaurus terms that are related to the users query. Usually specific kinds of relations are tested ("synonyms", "hyponyms", "hypernyms", "meronyms", etc).

In order to allow the query module to expand the query with related terms, the Query Expansion Web Service is employed. This Web Service has one method (getRelatedWords) that accepts one word, searches the word in the ontology and returns a list of related weighted words. The words are weighted according to the "semantic distance" to the original word. The number of retrieved results depends on the specific word: the list will be longer in case of words related to many terms, it could be empty otherwise.

The implemented system is composed of several logic modules, as shown in Figure 4. The main module is the Ontology Manager. Its aim is to act as an ontology repository, allowing query and modification of the ontology. The Query Expansion Web Service invokes the ontology manager for retrieving the word

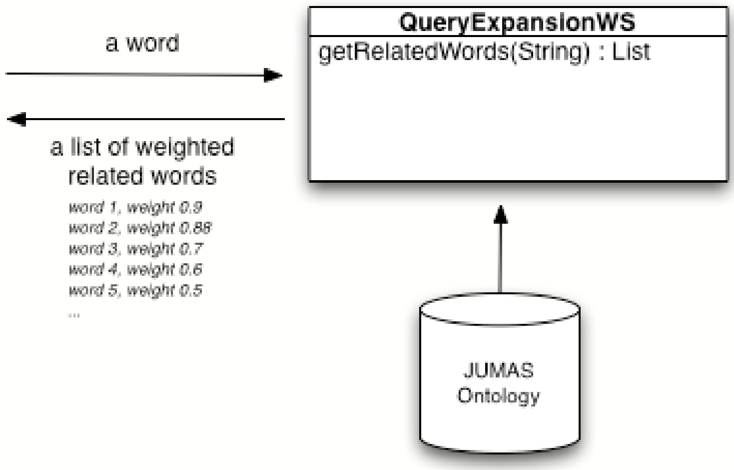


Fig. 3. An example of QE

to expand and calculate the similarity between the words. The web service can be invoked by external services and by the Query Expansion GUI. This GUI allows to the user to test the query expansion module before the integration with the other modules. Another GUI is implemented: the Ontology Management

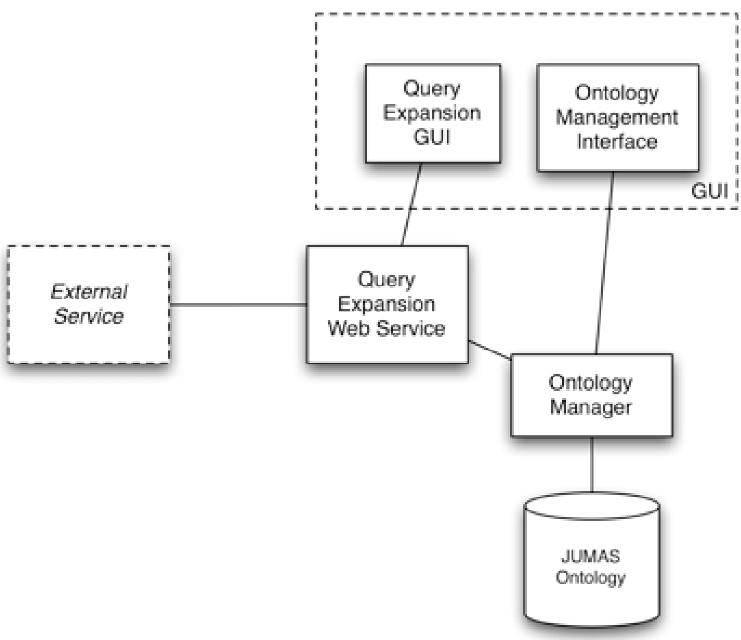


Fig. 4. Query Expansion Modules of the JUMAS system

Interface. It allows to navigate and edit the ontology, allowing create new word, deleting existing words, and the relations between the words. In the following paragraphs, the main modules will be described.

The query editing interface allows to modify the vocabulary, (see Figure 5), adding and removing terms and relations. Each modification performed by the user on the ontology is automatically saved in the ontology repository, used by the query expansion component. The interface is divided into four areas, each one regarding a different property. The properties are *Broader term*, *Narrower term*, *Related term* and *Synonym*.

In the figure the Italian word *cassaforte* (i.e. *strongbox* in English) is shown. The strongbox is a kind of box, so the word *contenitore* (i.e. *box*) it is a broad term of cassaforte. The related terms are *baule* (i.e. *trunk*), *cassa* (i.e. *till*), *combinazione* (i.e. *combination lock*), *grimaldello* (i.e. *lock picking*), and *serratura* (i.e. *lock*).

Synonyms are different words with very similar meanings. In the example the synonyms are *forziere* (i.e. *casket*) and *scrigno* (i.e. *treasure chest*).

To add a relation from the current term to another one, it is sufficient to write the term in one of the four text–fields and press the related button (e.g. to write a synonyms click the *Add synonym* button will be chosen). If the term is already in the ontology, a new relation between the two terms is added. In the term is not in it, it is automatically created. The interface displays the current relations. By clicking on the *delete* link near each relation, it is possible to remove the relation

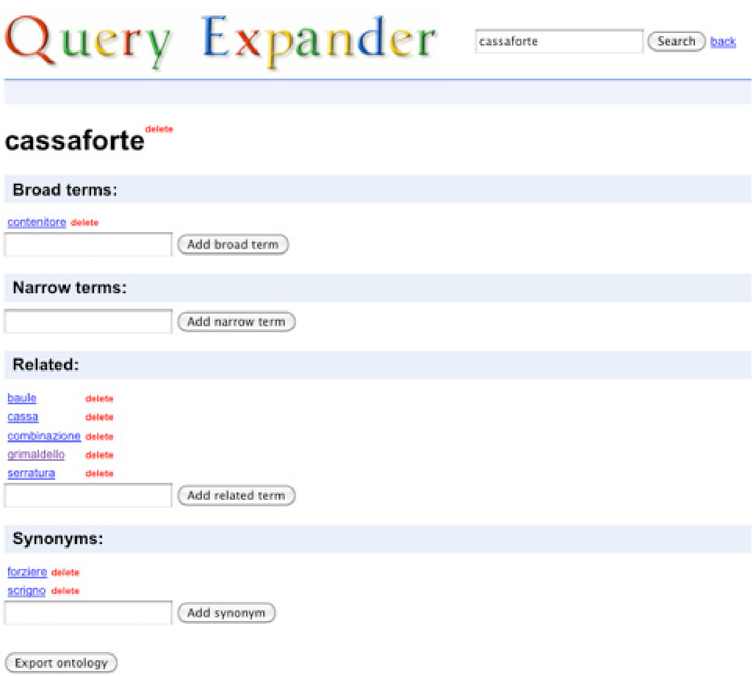


Fig. 5. A screenshot of the Ontology Management Interface

between the two words. When a word is removed from the ontology (by means of the *delete* link near the word name), all its relations are automatically deleted. Thanks to this interfaces the user can also export the ontology into an OWL file, clicking on the *Export Ontology* button.

An opportune GUI (see Figure 6) allows the user to test the query expansion module. It is very simple: when a user inserts a term, the page containing the expanded terms is generated. From the resulting page it is possible to navigate through the result clicking on the words. For each result it is presented the score (i.e. the level of similarity), the relation followed and the terms visited. For example, the similarity between *cassaforte* and *portagioielli* is 0.5; *cassaforte* is a synonymous of *scrigno* and *scrigno* is a synonymous of *portagioielli*. By clicking on the words, the user can finally access the editor interface. The similarity degree depends on the properties which are navigated. The weight of the properties can be changed through the configuration interface, accessible by clicking the *config* link. Through this interface it is possible to configure several parameters of the query expansion method, in particular:

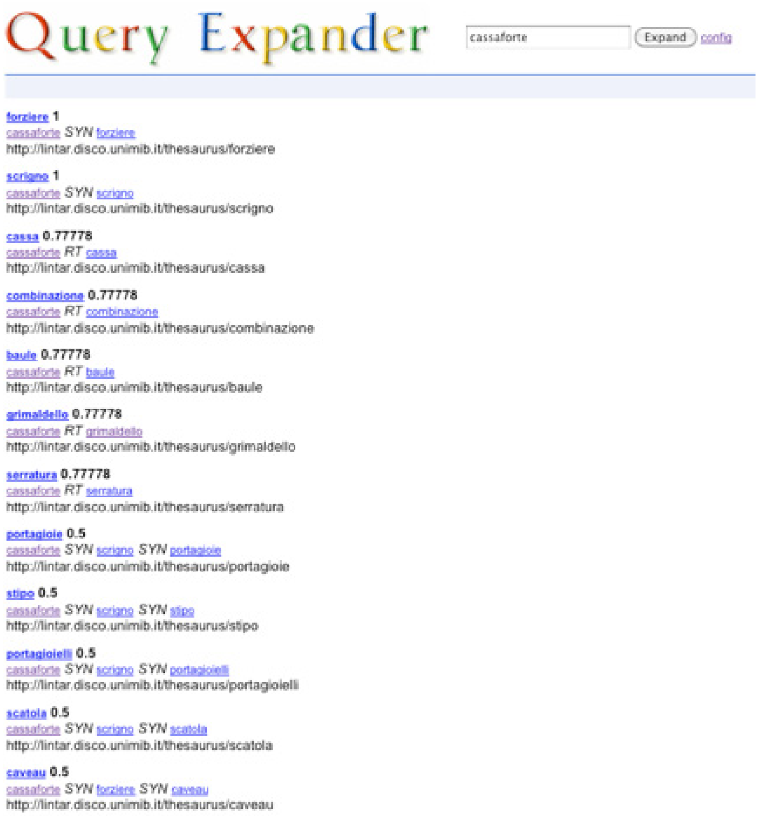


Fig. 6. Query Expansion GUI

- the max distance of the retrieved words;
- the weight for the different properties;
- the weight of the single word/multi words;
- the threshold;
- the max length of the multi-word results.

6 Conclusions

Ontological query expansion is a field in which a lot of improvements are theoretically possible. In this work we tried to set up an environment for testing and discovering potentiality of expansion through ontologies: we created a platform that allows to easily implement and analytically compare different methods and we started tests and comparisons to evaluate the usefulness of methods and techniques. The obtained results demonstrated that the platform works and can be useful to the goals for which it has been implemented. Moreover the results showed the possible scenarios in which the four methods implemented could obtain the best performances. We concluded our work by considering possible future developments in both the extension of the platform, the concrete applications and the researches that could be started from the results obtained.

Future work consists in the further development of the platform: to this aim, the first step is the evolution of the analyzed thesaurus of the current version towards a real and complete ontology: the exploitation of more complex relationships and properties should theoretically bring us to better results in QE, but the current development of research in this field is not supported by adequate practical advances.

In this sense, the platform has been tested in the context of the JUMAS project (*Judicial Management by Digital Libraries Semantics*), co–funded by the European Community (grant agreement number FP7-214306) in the context of the 7th Frame Programme. The platform will be used in JUMAS to implement a query expansion interface for users in the legal domain (e.g. judges, witnesses and so on) in order to help them to find significant audio–video transcriptions.

References

1. Allen, L.E., Saxon, C.S.: Achieving fluency in modernized and formalized hohfeld: Puzzles and games for the legal relations language. In: ICAIL, pp. 19–28 (1997)
2. Hamfelt, A.: Formalizing multiple interpretation of legal knowledge. Artif. Intell. Law 3(4), 221–265 (1995)
3. Valente, A., Breuker, J.: On-line: An architecture for modelling legal information. In: ICAIL, pp. 307–315 (1995)
4. McCarty, L.T.: A language for legal discourse i: Basic features. In: ICAIL, pp. 180–189 (1989)
5. Stamper, R.K.: The role of semantics in legal expert systems and legal reasoning. Ratio Juris 4(2), 219–244 (1991)

6. Breuker, J., Winkels, R.: Taking ontological commitments seriously: Folaw, a functional core ontology of legal knowledge. In: Hérin, D., Zighed, D.A. (eds.) EGC. Extraction des Connaissances et Apprentissage, vol. 1, pp. 9–25. Hermes Science Publications (2002)
7. Kralingen, R.W.V.: Frame-Based Conceptual Models of Statute Law. Kluwer Law Intl., Dordrecht (1995)
8. Breuker, J., Hoekstra, R.: Epistemology and ontology in core ontologies: Folaw and lri-core, two core ontologies for law. In: Proceedings of the EKAW 2004 Workshop on Core Ontologies in Ontology Engineering, Northamptonshire, UK, pp. 15–27 (2004)
9. Winkels, R., Boer, E., Hoekstra, R.: Clime: lessons learned in legal information serving. In: Proceedings of the European Conference on Artificial Intelligence (ECAI 2002) (2002)
10. van Engers, T.M., Vanlerberghe, R.A.W.: The power-light version: Improving legal quality under time pressure. In: Traunmüller, R., Lenk, K. (eds.) EGOV 2002. LNCS, vol. 2456, pp. 75–83. Springer, Heidelberg (2002)
11. Breuker, J.: The construction and use of ontologies of criminal law in the ecourt european project. In: Golaczynski, V. (ed.) Proceedings of Means of Electronic Communication in Court Administration (2003)
12. Tuominen, J., Kauppinen, T., Viljanen, K., Hyvnen, E.: Ontology-based query expansion widget for information retrieval. In: Proceedings of the 5th Workshop on Scripting and Development for the Semantic Web (SFSW 2009), 6th European Semantic Web Conference (ESWC 2009) (May 31-June 4, 2009)
13. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. 19(1), 1–27 (2001)
14. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR 1994: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 61–69. Springer, New York (1994)
15. Ruthven, I., Tombros, A., Jose, J.M.: A study on the use of summaries and summary-based query expansion for a question-answering task (2001)
16. Navigli, R., Velardi, P.: An analysis of ontology-based query expansion strategies. In: Proc. of Workshop on Adaptive Text Extraction and Mining (ATEM 2003), in the 14th European Conference on Machine Learning (ECML 2003), pp. 42–49 (2003)
17. Durão, F.A., Vanderlei, T.A., Almeida, E.S., de Meira, L.S.R.: Applying a semantic layer in a source code search tool. In: SAC 2008, pp. 1151–1157. ACM, New York (2008)
18. Xu, X., Zhu, W., Zhang, X., Hu, X., Yeol Song, I.: A comparison of local analysis, global analysis and ontology-based query expansion strategies for bio-medical literature search (2006)
19. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. Inf. Process. Manage. 43(4), 866–886 (2007)
20. Delort, J.Y.: A user-centered approach for evaluating query expansion methods. In: WI 2005: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, pp. 586–589. IEEE Computer Society, Los Alamitos (2005)
21. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (July 2008)
22. Tudhope, D., Alani, H., Jones, C.: Augmenting thesaurus relationships: Possibilities for retrieval. Journal of Digital Information 1, 15 (2001)

23. Alani, H., Jones, C., Tudhope, D.: Associative and spatial relationships in thesaurus-based retrieval. In: Borbinha, J.L., Baker, T. (eds.) ECDL 2000. LNCS, vol. 1923, pp. 45–58. Springer, Heidelberg (2000)
24. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms (1997)
25. Andreou, A.: Ontologies and query expansion. Master's thesis, School of Informatics, University of Edinburgh (2005)
26. Calegari, S., Pasi, G.: Personalized ontology-based query expansion. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 256–259 (2008)
27. Wielinga, B.J., Schreiber, A.T., Wielemaker, J., Sandberg, J.A.C.: From thesaurus to ontology. In: K-CAP 2001: Proceedings of the 1st International Conference on Knowledge Capture, pp. 194–201. ACM, New York (2001)

A Mereology-Based Ontology for Services Science: Example of an e-health Service Modelling

Florie Bugeaud[1,2] and Eddie Soulier[1]

[1] University of Technology of Troyes – ICD/Tech-CICO FRE CNRS 2848,
12 rue Marie Curie – BP2060, 10010 Troyes Cedex, France
{florie.bugeaud,eddie.soulier}@utt.fr
[2] Orange Labs – RD/BIZZ/BUC/RTS,
38-40 rue du General Leclerc, 92794 Issy-Les-Moulineaux, France
florie.bugeaud@orange-ftgroup.com

Abstract. This paper presents an ontological approach to formalize the concept of service upstream of the innovative process of telecommunication operators. These operators try to propose innovative solutions to meet their customers' expectations and to ensure their market position. We pursue the discussion of the SSME community and propose to conceive the service as a dynamic phenomenon: the "Service System". Traditional ontologies seem inadequate to represent it, so we introduce a new primitive (dynamic entities) based on the Mereology and the General Process Theory. A semi-formal method has been created through the import of key documents, the extraction of candidate-terms, the generation of an actions network (Galois lattice) and the construction of a mereological ontology. This method and its supporting tool OntoStoria² are currently tested within the process of opportunities research of a French telecom operator in order to promote the collaboration, sharing of knowledge and creativity of its innovators network.

Keywords: Services Science, Ontology, Mereology, Service System, Innovation.

1 Introduction: Innovation and Services Systems

The telecommunication operators have to identify and develop more and more innovative solutions. They try to meet a highly competitive context and the deep change of their customers' role and expectations [1]. But the evolution of ICT (Information and Communication Technologies) has led to important economic and social changes. It has significantly transformed the way of designing and delivering services. Moreover, these operators find in their value networks the skills and resources that are sometimes lacking in-house. But these networks also change their borders. Our work is interested in the upstream stage of the services innovation cycle of such an operator. Generally, telecom operators implement a design process which includes a step of opportunities research. During this step, their communities of innovators (i.e. innovation actors) have to interact in order to describe the service situation of the targeted customer / segment and then try to imagine some ideas of new IT or telecom services that fit the target lacks or expectations. In order to describe, evaluate and mature a

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 123–134, 2010.

situation, they implement a design reasoning related to a complex and collaborative practice in which they share and co-create knowledge [1][3]. But the implementation of this reasoning remains difficult because of the distributed and inter-professional (marketing, usage, ergonomics, and engineering) nature of their network, the lack of a structured approach and the inadequacy of the supporting tools. However the core problems are linked to the complexity of the concept of "service" that remains poorly understood and the the lack of interest for the notion of experience. The innovators meet some difficulties to imagine and represent the phenomenon of the service experience. And the customers meet some difficulties to determine the offered experience. One objective of our work is to address these difficulties by providing an integrated framework and a shared representation. This framework support their communication, collaboration and creativity upstream of the design cycle. The knowledge engineering provides some interesting elements. It facilitates the extraction, the annotation and the structuring of knowledge.

This article focuses on the conceptual representation of a service based on the information and knowledge which are available upstream of the innovation cycle. This requires a description of the concept of service at a business and more abstract level than the one we usually find in the literature (i.e. reusable technical functionality). We worked on the service theories and classifications (socio-economic, organizational, marketing and technology approaches that are rarely integrated) [2][3] and on the discussions of SSME (Services Science Management and Engineering, a multidisciplinary approach, initiated by IBM and several universities). Based on their proposal, we conceive the service as a "Service System" [4][5]. The Service System is an artefact, the object of a co-production between a client and a supplier. But it is also a dynamic configuration linked to the combination of heterogeneous entities. These entities form "a service" at a given moment and in a given situation. In other words, it is a dynamic and performative phenomenon related to a customer's help request to a supplier to transform a reality or a property. Then the question is: how to represent these kinds of informational collections of processes or procedural entities [6] which express a particular phenomenon (i.e. an experience)?

An important point of our work is related to the criticism of the substantiality principle which is the basis of all traditional ontologies. These ontologies use static and located entities and they consider the dynamics of these entities only as properties or secondary categories. We therefore propose a formal model, a construction method and a tool to explain and implement the information and knowledge surrounding the Service System that the innovators are studying. Indeed, we do not try to articulate the innovators' points of view. We propose them a more high level view of the overall service situation they have to study. It allows them to identify some tracks of new services by the modelling and simulation of dynamic situations.

This paper is organized as follows. Section 2 presents the arguments in favour of a new generation of ontology and the emergence of the proposed knowledge model to represent Services Systems. Section 3 describes OntoStoria[2]: a design and construction method of mereological ontologies, and its web-based tool. Section 4 gives the results of our first experiment within the e-health domain (i.e. modelling of the remote monitoring of diabetics patients). Section 5 concludes and points some perspectives.

2 Modelling of Dynamic Phenomena

2.1 Inadequacy of the Substance Epistemology

The service is often considered as a hierarchy of core services and peripheral services. The ontological representation of a Service System can be considered in order to help the telecom innovators to better conceptualize a customer's service situation as a dynamic phenomenon or a configuration of heterogeneous entities in a specific space-time. However, the traditional ontologies seem inadequate regarding the nature of the Service System. Ontology is the study of categories of things that exist or may exist in some domain [7]. Domain ontologies are based on an epistemology of the substance which expresses the being / thing / object in a nominal and static way. Even if some existing ontologies use events or processes as first class entities, domain ontologies describe the systems they study through the concept of object, the admission of the identity and object permanence principles, the possible consolidation into abstract classes and the instantiation as a redefinition of the object by the properties of the class. A process is only considered as "*the particular performance of a human activity, such as a particular running or reading, or, the particular occurrence of a 'subjectless', 'absolute', or 'pure process' : a snowing or thundering in a particular spatiotemporal region*" [8][9]. This substantial principle does not fit the dynamic and experiential nature of the Service System (collection of processes / dynamic entities). Moreover, the documents, that are used by telecom innovators to describe the services situations and interactions they address, are structured in a verbal form.

The representation of the "acting" service or the experience is therefore linked to the computation of the dimensions that impact its conceptualization as a process or a combination of parts in a given moment. Its elements can be considered as episodes or service tracks. We make the hypothesis that the reality is a continuous flow where "things" are processes. Their mode of existence is their activity mode and not their supposed universal substance. So we consider an ontological alternative based on dynamic categories rather than on abstract classes and static concepts. The role of ontologies in Information Systems seems to become central [10]. Domain ontologies provide some strategic improvements in terms of knowledge management (information research, knowledge sharing, etc.) [11]. However, in order to adopt a dynamic view answering the question of the service experience representation, we have studied the field of processes ontologies and mereological principles. Indeed we make a new hypothesis: the mereological "part-whole" relations allow getting the dynamic and systemic dimensions of a Service System when some innovation actors try to describe and understand it to imagine and propose some innovative solutions that will be adapted to precise spatio-temporal experiences.

2.2 Emergence of Process Ontologies

A new generation of ontologies seems to emerge: the processes ontology [6]. The conceptualization of dynamic categories is becoming possible thanks to a process-oriented metaphysics. It can be realized in a conventional way through aspectualities or in a less traditional way through an alternative theory to the paradigm of the substance [8]. According to our previous hypothesis, it is necessary to identify dynamic entities based on the mode of their activity and the kind of associated control (nature of the participants in the process / activity) rather than on objects (their properties, identity, information structure, etc.). It does not disprove the need for a knowledge theory of the area. But this knowledge will be expressed as dynamic categories rather than static categories. This implies to reverse the usually adopted perspective: from static entities to "development" / "progress" whose spatio-temporal structure and arrangement are complex. We use the basic principles of the Mereology and the General Process Theory to consider the modelling of dynamic categories and their associated situations. Indeed, a scientific track is to consider the "part_of" relationship rather than the traditional "is_a" relationship.

2.3 Contribution of the Mereology

The Mereology comes from philosophy. It is based on the formal ontological study of the "part-whole" relationship [13]. The semantic relation "part_of" (called "meronymy") represents a hierarchical and partitive relationship (e.g. the roof of a house). The main principles of the Mereology are the transitivity, non-reflexivity and anti-symmetry. The main kinds of meronymic relations can be established by the combination of three basic properties:

- Configuration: the parts support, or not, a structural or functional relationship with another individual or with the whole,
- Homeomery: the parts are, or are not, the same as the whole (same nature, same properties, etc.),
- Separability: the parts can, or can not, be separated from the whole (without denaturing the whole's identity).

Winston, Chaffin and Herrmann have identified, in 1987, six kinds of relationships (some of them are challenged by other authors): component-object, member-collection, portion-mass, material-object, phase-activity, area-zone [14]. Here we can note that some of them are more suited than others for the modelling of Services Systems: phase-activity ("script") and element-object (a service is an artefact and it is the object of a co-production). Some predicates have also been defined to describe these meronymic relationships: proper part, direct part, overlap, underlap, etc. Moreover, these relationships give rise to operations between entities such as sum, product or extension that can affect the parts of a whole.

2.4 Contribution of the General Process Theory

J. Seibt investigates for an ontological theory of emergence (i.e. conceptual claim based on empirical phenomena) [8]. She proposes a mereological extension which is non-standard and non-transitive. It is called the "General Process Theory" (GPT). This ontological framework for processes uses a new ontological category which is dynamic, concrete, non-particular (i.e. located in multiple and uncountable ways) and non-universal: the "general process". When processes interact or "interfere", they modify the representation of the complex process they form. In general, every verbal predication can be conceived as an activity in a general process, and may play different inferential roles depending on the context. GPT is based on:

- The mereological signature: homeomery and anhomeomery,
- A non-transitive "part-whole" relationship and some levels of partition (corresponding to the levels in the hierarchical tree and useful to identify and refer to entities / processes according to their position in a given partition)
- The principles of interference, mereological sum and product, spatio-temporal location.

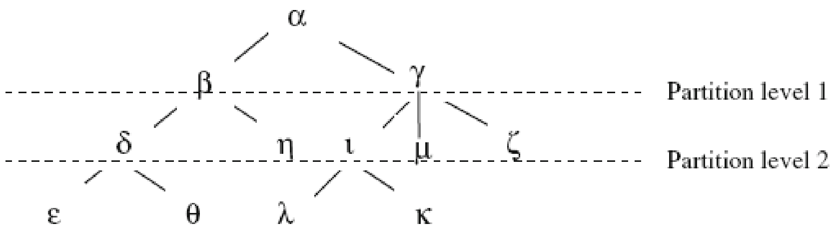


Fig. 1. Schematic representation of a mereological ontology in GPT [8]

The GPT non-transitivity principle changes the way of connecting some parts into a whole. A consequence is that the GPT reduces the number of possible relationships to the non-overlap and overlap. Another consequence is related to the operations of sum and product that can only be extended to the first partition level. Then any arbitrary amount of general processes can be created and any amount of general processes is itself a general process. Moreover, the "part_of" relationship does not necessarily involve a spatial or temporal inclusion. Processes can be arbitrarily dispersed in space and time. The GPT uses the definition of mereological properties, the analysis of mode of occurrence, the pattern classification of homeomery and the analysis of dynamic parameters in order to characterize each entity (see 3.1). It can thus answers the need of identifying the dynamic entities, their mode of activity, the nature of participants and the interactions between entities (causal, temporal, etc.). These framework helps us to capture the systemic and the dynamic dimensions of a targeted Service System (thanks to the definition and the representation of the “part-whole” relations and the interference/interactions between the processual entities of this Service System), then to imagine the possible scenarios / configurations and the service experiences to finally imagine adapted solutions.

3 OntoStoria[2]: Method and Tool

3.1 Methodology to Define and Build a Service System Mereological Ontology

We use the capabilities that are offered by the Mereology and the GPT to provide an adapted description of the Service System concept. The proposed method is based on a semi-formal semantic description of dynamic categories (see figure 2) (note that an ongoing work is related to the study of existing methodology for ontology engineering in order to improve and implement our proposition):

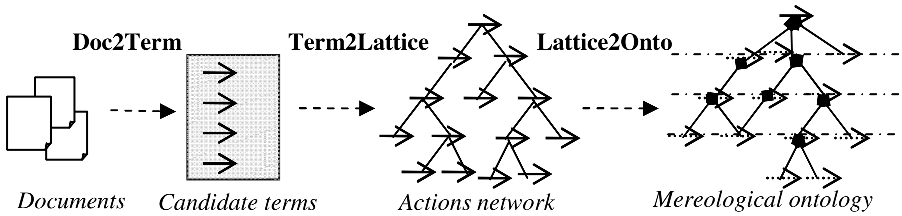


Fig. 2. The OntoStoria² method

The first step (*Doc2Term*) concerns the extraction of key terms from the documents which are available upstream of the design cycle of telecom operators (see 3.2 for the mechanisms). The constitution of a corpus of documents about the studied Service System is an important step. These documents contain stabilized knowledge coming from previous studies, information of the market, specific documentation of the studied domain (e.g. e-heath), etc. The extraction of key terms helps formalizing the elements that were previously informal. Moreover, the innovators, who participate in the co-construction of the Service System representation, can modify the resulting key terms structure (i.e. delete some terms or adding their own knowledge). The following steps are completely transparent for the user (until the display of the result).

The second step (*Term2Lattice*) is linked to the hypothesis that we all are accustomed to the paradigm of the substance (see 2.1). Therefore, we need to move from the conceptual space to a dynamic space [7][8]. Saillot, Poitrenaud and others authors consider that when we think about an object, we must grasp not only the semantic dimension but also the pragmatic dimension [19]. There is a direct relationship between objects and activities. This can be used to go beyond the mental habituation we introduced. It provides a direct access to the processes world and therefore to the pragmatic. The Galois lattices are helpful to represent hierarchical networks of categories (i.e. semantic networks of actions). They use this basic combination between actions (procedural knowledge) and objects (declarative knowledge). For us, the benefit of such a network is to structure the key terms from *Doc2Term*, and then to consider a processual perspective. The Galois lattice generation is based on the following rules:

- The triplet (O, A, I): a set of objects O, a set of attributes A (i.e. actions which are applicable to the objects) and the binary relations I between O and A (Rule 1: $I \subseteq OxA$).
- Two "Galois connections": An intension of Oz (Rule 2.1: f(Oz)=An, i.e. all the subsets of A with which the objects of Oz have a common binary relation) and Oj extension of As (Rule 2.2: f(As)=Oj, i.e. all the subsets of O with which the actions of As have a common binary relationship).
- The reciprocity of the Galois connections: Oi is the extension of Ai and Ai is the intention of Oi (Rule 3: g(Ai)=Oi and f(Oi)=Ai). Then (Oi, Ai) is a "concept".
- The order relation: the set of concepts is ordered thanks to the inferiority (denoted $\leq$) and the inclusion (denoted $\subseteq$) relations (Rule 4: $(Oi,Ai)\leq(On,An)$ if $Oi \subseteq An$ or $Ai \subseteq An$).

We have created four algorithms based on these rules in order to transform the entities extracted from the upstream documents into an actions network (see 3.2).

The resulting actions network helps us to identify "candidate processes". Then, the next step (*Lattice2Onto*) determines the ontological links between thse entities (i.e. the candidate processes) and builds a complete representation. Here we use some mereological conditions and criteria which have been proposed by the General Process Theory [8]. The process typology is based on the:

- Classical criteria: dynamicity, unboundedness, distributivity, homeomerity,
- Mode of occurrence criteria: completeness, resumability, recurrence,
- Mereological criteria: homomerity and automerity pattern (according to spatial and temporal occurrence),
- GPT Classification criteria:
  - Participant structure (according to number and type of causal agents and patients),
  - Dynamic constitution (according to process architecture, e.g. sequences, forks, joints, cycles, etc.),
  - Dynamic shape (according to their trajectories, verbal aspects),
  - Dynamic context (according to its influence on the generative environment of the process).

Thanks to these criteria, we obtain several kinds of entities' profiles: activity, event, thing, substance and quality. In the case of Services Systems, the activity and event profiles seem relevant. We then look at the relationships between these entities, the number of partition levels (i.e. levels in the tree) and the kinds of involved dynamics. Three main characteristics have to be taken into account:

- Profile of each entity,
- Homeomy ("like-partedness") and automery ("self-partedness") degrees,
- Type of relationship: non-overlap or overlap,
- Type of operations: sum, product, cause.

The discovery of part-whole relationships leads to build a mereological ontology. This framework allows the emergence of various types of dynamics (sequential, co-occurring, causal) when it is applied to a particular field. The final ontology is thus very useful to represent the possible dynamic configurations of a Service System.

3.2 Proposition of a Web-Based Design Studio

We are implementing a Services Systems Design Studio based on the previous method. It is a web-based tool associated with a database server (see figure 3), a design space (to create or open project), a toolbox and a collaborative space. It can be used in an asynchronous way (through the remote and inter-professional network of innovators) or in a direct access (an innovator or a group of innovators during a brainstorming). It uses the traditional mechanisms of social networks for the asynchronous access: profiles, tags, etc. In both cases, the first step is related to the initialization of the theme / topic by an innovator or a group of innovators that either received a specific request for a specific service situation (e.g. a demand coming from the direction), or think that a study of this service situation may lead to the discovery of new telecom services. Various mechanisms are then implemented (see below).

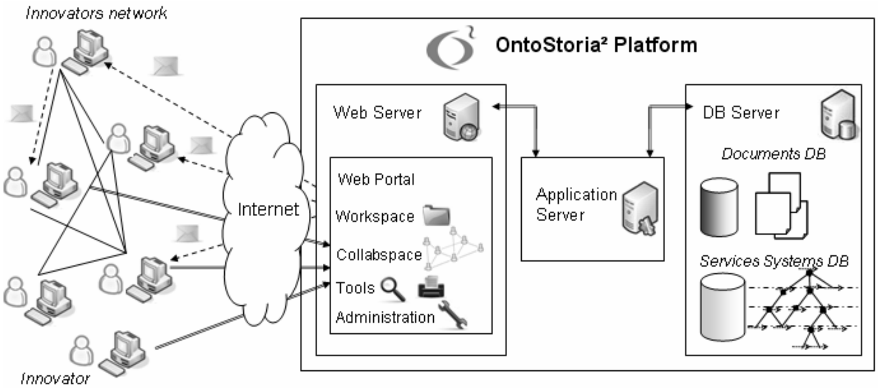


Fig. 3. Global architecture of the web-based studio OntoStoria²

The Natural Language Processing (NLP) and Text Mining propose some mechanisms to create indexes or build ontologies from texts. The extraction of key terms from marketing or R&D documents requires the definition of an analysis framework and some rules for the procedure of terms research and extraction [15]. Here, the "candidate terms" are the key objects and actions of the studied Service System. The main steps of *Doc2Term* (see figure 2) are:

- Locating and cutting phrases / terms [16],
- Identifying relationships / correspondences between them,
- Identifying and eliminating redundancies and synonyms,
- Extracting and displaying terms in a list-like structure.

This list is then provided to the innovators (considered as human analysts [15]) who participate in the Service System study. They may remove a term they consider improper, add a missing term and validate the selection of "candidate terms". Based on the triplets objects/actions/relationships, the four algorithms (related to the Galois Lattice rules, page 6) create a hierarchical representation (*Term2Lattice*):

- Rule 1: determining the possible sets of actions and sets of objects,
- Rules 2.1 and 2.2: researching the objects / actions pairs,
- Rule 3: identifying the objects / actions pairs which are concepts,
- Rule 4: classifying the concepts thanks to the actions order relation (because we are here interested in the actions: $(Oi, Ai) \leq (On, An)$ if $Ai \subseteq An$).

The actions network is then generated. Its conversion into a mereological ontology (*Lattice2Onto*) requires two main steps:

- Describing the entities profiles (activity or event) (see 3.1 for the typology),
- Identifying the part-whole relationships between processes (sum, product, etc.) and characterize their nature.

The resulting list of entities profiles and interferences as well as the focus of interference (i.e. the macro-process / phenomenon / main dynamic we want to study) allow us to fit all processes and determine the number of partition levels in the graph. We obtain a complete description of everything that can intervene in the procedural universe of the studied Service System.

4 Experiment on e-health Service System

We applied the proposed method and calculations to an e-health Service System: the remote monitoring of diabetics patients. It is supported by a telecom solution (PDA + Internet) and has been the object of several experiments a few years ago [17]. But we have discovered that, during these experiments, the telecom innovators that have worked on the diabetics monitoring description (in order to imagine some technical solutions to support it) met some coordination difficulties and did not converge towards a consensual representation. It is thus a good example and we have decided to position ourselves “as innovators” during an opportunities research process and to test our proposition (a more complete evaluation with a “real” group of innovators is currently under preparation). We used different kinds of documents as input: sales brochures, press articles and conferences presentations. The key objects and actions of the remote monitoring of diabetics have been extracted from this corpus. As the result was sizeable, we only treated a sample in this first experiment:

- Objects: medical advice, consultation, evolution prevention, constraints, events, emotions, weekly remote monitoring, blood sugar level balance, consulting at the doctor’s office, self-monitoring rhythm,
- Actions: sending the data, advocating a change in treatment, recognizing a feeling of faintness, receiving a medical advice, adding some information, using lancing, visualizing the results, calculating the blood glucose rate, planning the meals for the week, making an injection.

The four “Galois-lattice-based” algorithms have been applied to the sample triplets.

To avoid overloading this paper, the figure 4 proposes a reduced view of the resulting actions network. We have removed the objects and actions redundancy in each node in order to highlight their specificity. The inheritance and inclusion mechanisms can help to rebuild their extensions and intensions.

According to Saillot *et al.*, the cognitive complexity is the measure of quantitative parameters in the graph [18]. If we analyze the figure 4, we find 8 categories (nodes), 8 relations (arcs), 4 levels of abstraction (maximum number of class on a line), 2 categories that have only one immediate neighbour, and 6 categories that have more than one immediate neighbour. Moreover, the "superordinate category" (i.e. whose source is remote and which has an important extension / number of objects) is the node at the top (its main action is "advocate a change in treatment"). It corresponds to the main representation of remote monitoring of diabetics in people’s mind. The "subordinate categories" (i.e. whose sources are close and which have an important intension / number of actions) are the two nodes at the bottom (their main actions are: "use a lancing device" and "recognize a feeling of faintness"). They are probably a later representation of the disease monitoring in people’s mind.

Then, we have applied the mereological ontology construction steps to our case. The profile of each process of the sample has been identified thanks to the process typology and the part-whole relations have been detected. For example:

- "visualizing the results" meets the criteria of the activity profile,
- "adding some information" meets the criteria of the activity profile,
- "sending the data" meets the criteria of the event profile,

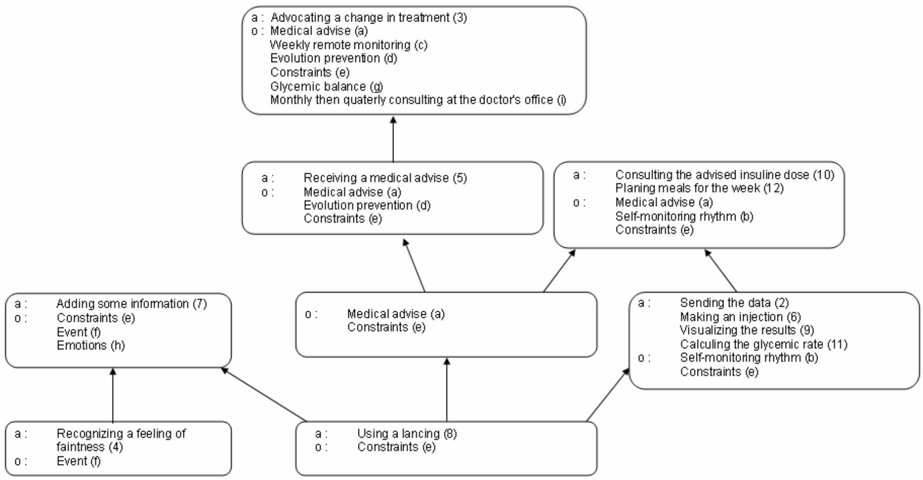


Fig. 4. Actions network on the sample

- the interference between "visualizing the results" and "sending the data" is a part-whole relationship,
- the interference between "adding some information" and "sending the data" is a part-whole relationship, etc.

Once all processes and interferences have been qualified, their dependencies help us to generate the ontological model (see figure 5 on the original sample).

In this first model, the interference focus corresponds to the process of "Diabetes remote-monitoring" and there are three levels in the partition. A more complete view is under construction. It will take into account and show the other interactions between entities (e.g. functional, causal, etc.) at the same partition level. This is a key

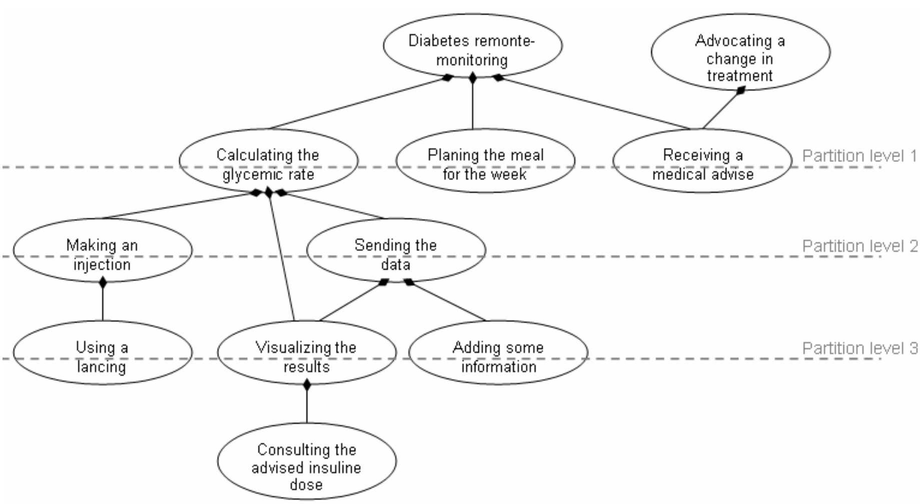


Fig. 5. Mereological ontology on the sample

point that will help us to consolidate the discovery of the possible scenarios / configurations and therefore experiences in a specific service situation. Indeed, it will help the innovators, during the opportunities research step, to not only better conceptualize and understand the addressed service situation as a dynamic configuration of heterogeneous entities in a specific spatio-temporal region, but also to identify new ideas of telecom or IT services to answer the situation's lacks or customer's expectations (e.g. a classical voice server for the monitoring of older diabetics patients who are familiar with the PDA and web technologies).

5 Discussion

This paper is concerned with "what does" or "how interacts" what we talk about and the mereological relationships between the associated processes. We adopt a critical position against the paradigm of the substance and we pursue the discussions about processes ontologies. We propose the construction of mereological ontologies by categorizing each entity and each interference between these entities. Their part-whole relationships and their dynamics animate this procedural universe through all the possible scenarios. It therefore aims at supporting the telecoms innovators' collaboration and creativity through a shared representation of the services situations.

One of the main perspectives of our work concerns the complete experimentation of the method and algorithms on the remote monitoring of diabetic patients (this paper presents a sample) and on other Services Systems in order to verify the suitability of the GPT and OntoStoria[2] framework as well as the rest of the platform. Ongoing works are related to the comparison of existing initiatives in the field of events or processes ontologies (even if they are based on the paradigm of the substance) with our proposition and on the existing methodologies for ontology engineering. Another perspective is related to the representation of the mereological ontologies. There are still very few mereological representations. Some works about the mereotopology [19][20] propose a representation based on a logical description of the SEP triplets. These triplets are typically used to emulate the property of transitivity in the general theory. The GPT proposes a "part-whole tree" which is organized according to different levels of partition [8]. Our work is closer to the GPT proposal which does not use the property of transitivity in the decomposition of a whole into parts and does use the concept of interference as a key point. Therefore, the representation as a "part-whole tree" seems more appropriate. But in the context of telecom innovation, this kind of model can be hardly understood by some professions. It now seems necessary to consider the creation of more suited views for both innovators and customers (in both cases, we can imagine a kind of structure of dynamic trajectories in the mental space, one being related to the design of an offer and the other to an experience).

To conclude, this work is included in a more global approach. This approach describes and simulates Services Systems upstream of the innovation cycle of telecom operators. The proposed steps, in this paper, are part of a unique phase of cognitive representation of Services Systems. Other phases complete it: behavioral representation, computation of the service experience, simulation / animation, validation and detection of ideas of new services (that can be transferred towards the design and development phases of the innovation cycle). This approach and its main mechanisms have been or will be the object of other publications.

References

1. Bugeaud, F., Soulier, E.: Raisonnement de conception de services à l'aide d'un réseau social. In: 20èmes Journées Francophones D'Ingénierie des Connaissances (IC 2009), Hammamet, Tunisie (2009)
2. Tannery, F.: Le management stratégique des services: synthèse bibliographique et repérage des questions génériques. Finance Contrôle Stratégie 4(2), 215–259 (2001)
3. Bugeaud, F., Soulier, E.: Services Systems to Leverage Innovators' Knowledge: the Telecoms Industry Case. In: 10th IFIP Working Conference on Virtual Enterprises (PROVE 2009), Thessaloniki, October 07-09 (2009)
4. Spohrer, J., Maglio, P., Bailey, J., Gruhl, D.: Steps Toward a Science of Service Systems. IEEE Computer 40(1), 71–77 (2007)
5. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M.: Value-based Service Modeling and Design: Toward Unified View of Services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 410–424. Springer, Heidelberg (2009)
6. Soulier, E.: Storytelling, Plateformes Sociales et Ontologies de Processus pour la simulation du Mouvement. Habilitation à Diriger Des Recherches (2009)
7. Sowa, J.F.: Building, Sharing, and Merging Ontologies (2001)(unpublished paper)
8. Seibt, J.: Forms of Emergent interaction in General Process Theory. Synthese 166, 479–512 (2009)
9. Seibt, J.: Free process theory: Towards a typology of occurrings. Axiomathes 14, 23–55 (2004)
10. Guarino, N.: Formal Ontology and Information Systems. In: International Conference on Formal Ontology in Information Systems (FOIS 1998), Trento, Italy (1998)
11. Gandon, F.: Ontology Engineering: A Survey and a Return on Experience, INRIA Report no. 4396 (2002)
12. DAML-S Coalition: Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila., Martin, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T., Sycara, K., Zeng, H.: DAML-S: Semantic Markup ForWeb Services. In: Semantic Web Working Symposium (2001)
13. Varzi, A.C.: Standford Encyclopedia of Philosophy Mereology (2003 and 2009), `http://plato.stanford.edu/entries/mereology/`
14. Winston, M.E., Chaffin, R., Herrmann, D.: A Taxonomy of Part-Whole Relations. Cognitive Science 11(4) (1987)
15. Aussenac-Gilles, N., Bourigault, D., Teulier, R.: Analyse comparative de corpus: cas de l'ingénierie des connaissances. In: 14èmes Journées Francophones d'Ingénierie des Connaissances (IC 2003), Laval (2003)
16. Bourigault, D.: Conception et exploitation d'un logiciel d'extraction de termes: problèmes théoriques et méthodologiques. In: Clas, A., Thoiron, P., Béjoint, H. (eds.) AUPELF-UREF, Montréal, Traduction, Lyon. 4èmes Journées scientifiques du réseau thématique Lexicologie, Terminologie, pp. 137–145 (1996)
17. Franc, S., Dardari, D., Boucherie, B., Riveline, J.P., Biedzinski, M., Petit, C., Requeda, E., Leurent, P., Varroud-Vial, M., Hochberg, G., Charpentier, G.: Diabeo: an innovative Telemedicine system for the follow-up of type 1 diabetic patients. In: DTM, San Francisco (2009)
18. Saillot, I., Patou-Mathis, M., Richard, J.F., Sander, E., Poitrenaud, S.: Modéliser les activités cognitives des Hommes au Paléolithique. Math. & Sci. Hum. (159), 55–72 (2002)
19. Varzi, A.C.: Parts, wholes, and part-whole relations: the prospects of mereotopology. Data and Knowledge Engineering 20, 259–286 (1996)
20. Schulz, S., Hahn, U.: Mereotopological Reasoning about Parts and (W) Holes in Bio-Ontologiest. In: Conference FOIS 2001, Ogunquit, Maine, USA, pp. 210–221 (2001)

On Using the REA Enterprise Ontology as a Foundation for Service System Representations

Miguel-Angel Sicilia[1] and Manuel Mora[2]

[1] Computer Science Dept., University of Alcalá
Ctra. Barcelona km. 33.6 – 28871 Alcalá de Henares (Madrid), Spain
msicilia@uah.es
[2] Department of Information Systems, Autonomous University of Aguascalientes, Aguascalientes, Mexico, 20000
mmora@securenym.net

Abstract. The complexity of service systems has raised interest in developing ontologies that contribute to the conceptual clarification of value co-creation activities and that eventually serve as knowledge representations for systems exploiting reasoning and inference on those ontologies for various practical purposes. The Resource-Event-Agent (REA) enterprise ontology shares similar aims even though it addresses the more generic domain of economic phenomena as modeled by interchange events between agents. In consequence, the REA ontology appears as a good candidate to serve as a framework for a module dealing with an event based account of service provision that could be part of a broader service system ontology. This paper reports a preliminary analysis of the applicability of the REA enterprise ontology to the service domain, according to its formulation by Geerts and McCarthy [14] [15]. Service provision can be modeled as a kind of REA *transfer*, and the selling of a service that will be provided in the future can be modeled by associating the selling with the *commitment* to provide the service associated with the transfer. Different kind of *resources* can be used to model intangibles that are put into action during the service encounter itself, and a separate event structure can be used to model the service encounter. The paper sketches a preliminary refinement of the REA ontology to explicitly account for these aspects.

Keywords: services, service systems, service science, ontology, REA enterprise ontology.

1 Introduction

Service Science, Management and Engineering (SSME) is an emerging interdisciplinary academic and professional discipline, focusing on the design and management of co-generative service systems [8]. As services are pervasive as the "front stage" in economic activities of any kind [22], the role of software systems adequately supporting services is critical in approaches to services enabled by Information Technology (IT) infrastructure. This is in addition becoming more important with the widespread use of the Internet for e-commerce and other activities based on the Web [7], [11].

M.-A. Sicilia, C. Kop, and F. Sartori (Eds.): ONTOSE 2010, LNBIP 62, pp. 135–147, 2010.

The nature of services had raised interest in the development of knowledge models able to capture the inherent complexity of service systems. The approach to knowledge modeling known as *ontology engineering* appears as a promising vehicle for that task as shared ontologies aim at avoiding conceptual ambiguities, promoting reutilization and standardization, and serving as building blocks for more complex automated-reasoning systems [6], [16]. Some specific ontology initiatives have been reported to date that address the specificities of services, with *OntoServSys* ([21], to appear) as the one described in more detail and the unique to date that explicitly and at the same time exhaustively addresses all the key theoretical elements of SSME as an emerging discipline. Other relevant previous works are the reports from Ferrario and Guarino [12] and Ferrario, Guarino and Fernández-Barrera [13].

However, to the best of our knowledge there are no existing reports that attempt to evaluate existing ontologies as candidate building blocks to be reused for service ontologies. Among these existing previous models, the evolution of the Resource-Event-Action (REA) framework [19] into a full ontology appears to be an important precedent for modeling services, as it is accounting for value transfer in economic phenomena, which is also the key element in service from the business perspective. However, most examples found in REA-related research are related to good selling, and the specific elements of services are not dealt with explicitly (although in principle they are not excluded from the scope of the REA model).

More concretely, we consider here the REA Enterprise Information System (REA-EIS or REA enterprise) ontology, which is an evolution of the REA model for transaction processing of economic phenomena in a shared data environment [14], [15]. The REA-EIS neither provides an explicit service orientation nor includes a differentiation of services from selling of goods that has a clear reflection in the resulting models. This has leaded us to examine its applicability for modeling service systems and to look for necessary extensions or refinements that could be needed for that particular purpose. This paper reports on a preliminary exploration of the REA-EIS as described by Geerts and McCarthy [14] [15], and suggests extensions or refinements to integrate the sound conceptual backbone of REA with the specificities of service systems. It should be noted that there is an ongoing development of a new version of the REA ontology called REA2[1] that has not been fully considered here, but that might result in differences that also affect the extensions and refinements described hereinafter.

The rest of this paper is structured as follows. Section 2 briefly sketches the different views on service ontologies that can be found in previous work and discusses our focus in this paper. Then, Section 3 reviews the main elements of the REA enterprise ontology pointing out to elements requiring conceptual classification, extension or further specification to meet the needs of service systems modeling. Finally, conclusions and outlook are provided in Section 4.

2 The Different Views in Service-Related Ontologies

Engineering an ontology for services poses a significant challenge as the concept of service is understood differently depending on the discipline or application area. Alter [1]

[1] The REAv2-WIKI can be reached through this Web page:
http://www.managementinformatics.ugent.be/

has analyzed this disparity comparing the views of marketing, operations and computer science, resulting in three models: work system, value chain and lifecycle. In a different direction, the European project OBELIX[2] described an RDF service ontology in its deliverable D6.1 titled "Service ontology specification”. That model divided the ontology in three viewpoints: service value (the customer perspective), service offering (the supplier perspective) and service process (how the service is actually performed). Ferrario and Guarino [12] emphasize commitment of agents to provide the service, and elaborate with examples of public service, which entails a different perspective from other services in which value transfer is not mediated by taxation but it is explicit in a transfer of value. They clearly differentiate commitment, service bundling, acquisition, process and value exchange as interrelated but distinct kinds of service activities.

This plurality of views, aspects or models points out to the need of having a collection of ontologies covering different viewpoints rather than a single, larger ontology. In addition to these aspects, there are existing product and service classifications as eClassOWL [17] that serve as intensional catalogues of types of services (i.e. different service *contents*). However, in its current version, eClassOWL is a “product or service” ontology that merges products and services in the same taxonomy, as reflected for example in concepts as `ProductOrService`, `ActualProductOrServiceInstance`.

The term service is also used in the domain of Information Technology to refer to a concrete way of structuring distributed software systems. Service-Oriented Architecture (SOA) is an architectural style that supports service orientation, i.e. structuring software interfaces as distributed services that can be invoked remotely to obtain some effect or output data for some given input data. Obviously, these “software services” are not necessarily related to services in the economic sense, considered as value co-production activities. However, SOA has promoted a particular paradigm of structuring computer-based systems that fits well services as a human activity, and it has even fostered the business model of “Software as a service” (SaaS), in which a provider licenses an application to customers for use as a service on demand [5]. Also, the Open Group has recently released a draft for a SOA ontology expressed in OWL-DL[3], which could eventually be reused to match general service ontologies to software service ontologies. We should highlight that we are not dealing here with software service ontologies, so in the discussion below service is referring to events involving humans (directly or indirectly).

OntoServSys ([21], to appear) provides a comprehensive account of service concerns, based on an analysis of the literature regarding SSME. Among many other aspects, it covers actual service interaction including actors, resources and events. These form the basis of the operational model of any service system ontology, as *service encounters* are the target of design and management that can be considered to conform the core conceptual structure. That part of the ontology has a precedent in the REA framework [19], which provided a model strongly rooted in accounting and economic theory and addressed the issue of what phenomena should be captured in an enterprise system in general.

As mentioned above, services are considered essentially as commitments by Ferrario and Guarino [12]. This is connected also with the concept of commitment in the

[2] http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=61165
[3] http://www.opengroup.org/projects/soa-ontolo gy/

REA model, and there are several links between REA-EIS elements and the general framework of Ferrario and Guarino. These authors have proposed a general ontological foundation for the notion of service, which in turn is based on the notion of commitment. Their definition of service as "*A service is present at a time T and location L iff, at time T, an agent is explicitly committed to guarantee the execution of some type of action at location L, on the occurrence of a certain triggering event, in the interest of another agent and upon prior agreement, in a certain way.*" . This notion of service is related to a possibility of action, not the real "service encounter", and other authors refer to this notion as "service offering". The notion of commitment appears in the REA-EIS ontology, but the central element in REA models is the transfer of value. Even though the transfer is considered in Ferrario and Guarino framework [12] as part of "Service value exchange", that part of the model is not described to the level of detail that can be found in REA. In consequence, it seems reasonable that REA models can be combined with other service areas but still reused as a mature model.

In this paper, we are examining the position of the REA model which is rooted on accountancy and as such, focuses on value transfer. However, the theoretical standpoint of any ontology of services requires some basic ontological commitments that characterize services as opposed to goods. We approach services here as *events* (similarly to Ferrario and Guarino), but focusing on service as "*value co-creation through inter-action*". That notion can be integrated with good production and selling using theoretical economic frameworks. For example, let's take praxeology – the "science of human action" according to [20] – as a departure foundation. Then, we can consider the axiom that humans engage in conscious actions toward chosen goals as they have values (not necessarily rational) and they believe to have the technological knowledge that certain means will achieve his desired ends. In consequence, agents purposefully produce offerings and are served by the offerings of other agents that they select to achieve their aims. This can be combined with the axiom that means employed are scarce in relation to desired ends, and we end up with reflecting on the means to fulfill needs, and how humans organize themselves in production processes to help others in that fulfilling. This leads to the following categorization:

- Production and provision as *two separated actions*. This is the case of good selling, in which production is organized to produce tangible goods that are eventually stocked and later subject to selling. That last selling step could be considered a service also, but restricted to facilitating the transfer of the tangible item previously produced.
- Production and provision when *combined in the same, inseparable course of action*. This is the case of services, in which the value is created at request of the customer, with some degree of interaction with him and not necessarily involving any transfer of ownership.

In both cases, it is possible that the transfer of value is done in different arrangements, e.g. advanced payment or delayed payment. Both kind of combinations (production and service) can also be bundled together in complex ways. Combined production and provision seems to be the main aspect of service definition, so that value transfer is done in interaction with the customer. It should be noted that interaction does not necessarily mean face-to-face or physical interaction (as in services provided through the Web), and that there is not a need that interactions of customers and providers are not necessarily synchronous in time (as in the services of a private detective agency, in which they act on behalf of the customer).

These are the main elements considered as essential aspects of services here:

- *Inseparability*. Events cannot be inventoried, as they are created and consumed at the same time.
- *Intangibility*. As services are events, we cannot possess events (even thus we can own they results).
- *Perishability*. It is also a consequence of considering services as events. The lifespan of the service is that of the event itself, while in goods, these can be persistent in time.

These characteristics will be revisited in the following discussion as a test to the account of service presented. Heterogeneity and personalization are also usually mentioned as distinguishing characteristics of services from goods. However, this is not a clear distinction, as goods production has evolved in some cases to reach high levels of personalization, as it is common in “mass customization” approaches [9].

The REA model and its subsequent refinements intend to cover value transfer phenomena. In consequence, it is apparent that it should be able to cover service provision as interaction events. It should be noted this does not exhaust all the aspects identified previously by Mora et al. [21] and Ferrario and Guarino [12], but touches the key aspect of modeling the value transfer event structure.

3 The REA Ontology and Service Representation

The REA (*Resource-Event-Agent*) enterprise ontology is a core framework for transaction processing of economic phenomena in a shared data environment [14] [15]. As such, it is obviously of potential usefulness as a model for service interaction. However, there is a need to carefully analyze the ontology to assess if the specifics of service interactions are explicitly captured or specializations of some concepts or relationships are required.

Services represent the “front stage” of virtually any kind of business [22], when interaction with the customer takes place. Such interaction is sustained in many cases by some complex “back stage” machinery, however such back stage mechanisms are out of scope of our present effort.

In what follows, the main elements of the REA model are discussed in the aspects that are directly relevant to services. It should be noted that the good/services sector separation is nowadays considered largely artificial, so that a good model should account for a model of interaction in which both can be represented together.

Economic Events and Resource Exchange and Use

The REA model is organized around (composite) `Events`[4] in which `Agents` participate. Concretely, the definition of `EconomicEvent` in the OWL version of the REA

[4] In what follows, ontology elements are in Courier font, to differentiate them easily when reading the text.

ontology[5] is as follows: "*Represent either an increment or a decrement in the value of economic resources that are under control of the enterprise*". Concretely, there are "increment" economic events, and "decrement" economic events, which are related to each other via the predicate `duality` that "keeps track of which resources were exchanged for which others". In principle, this same distinction can be applied to services. The increments are related to receipt of money or other kind of resource, and the decrements in the case of good selling are the transfer of the tangible good (so, the model has the perspective of the provider). In the case of services, decrements are related to the capability of the provider to do service, e.g. when a service is being provided by a masseur to a customer, it cannot be simultaneously provided to other. This fits well with the notion of considering opportunity costs as a kind of economic resource. However, this also requires a very broad notion of "resources". For example, in OpenCyc[6] we can find the predicate:

```
(resourceAvailable AGENT RESOURCE)
```

Which means that `AGENT` has access to the object `RESOURCE` in such a way that `AGENT` could make some immediate use of `RESOURCE` in any type of action which required it. Resources are considered as instances of the OpenCyc concept `SomethingExisting`, which includes tangible things (as persons or machines) but also some intangibles as agreements or obligations. Resource unavailability for the time of the service (in the sense that cannot be used for other customer at the same time) is a way to model the outflow of the service providing part. This would require a ternary relation as for example `(resourceAvailable AGENT RESOURCE TIMEINTERVAL)`, or defining an additional concept representing time bound usage of resources as a specific kind of resource, e.g. the usage of an infrared sauna (an economic resource) for 15 minutes at some given time. This is a consequence of the intangibility of services, as what is transferred is the work of some resources for some time.

This in OWL could be expressed as a constraint on a `TemporalUseOf EconomicResource` concept in the form:

```
EconomicResource and temporalConstraint some TimeInterval.
```

However, this would entail some change in the model so that unavailability is temporally marked, and it is not clear that this matches the current semantics of REA that appear to be related to transfer of possession (but the formal OWL definitions are not precluding that matching). This would have the benefit to connect to models of availability management which are common in the practice of service planning.

The `EconomicAgents` relate to `EconomicEvents` via `participate` relations. They contribute `EconomicResources` they own to the event, which results in a "`stock-flow`", including an "`inflow`" and an "`outflow`". These elements are in principle applicable to service provision, provided that the stock flows include intangible elements. For example, in consultancy the output stock can be considered to be

[5] http://www.managementinformatics.ugent.be/REAontology/OWLdoc/

[6] OpenCyc is the open source version of the Cyc Knowledge Base (Lenat, 1995), which contains over one hundred thousands atomic terms, and is provided with an associated efficient inference engine.

the increase in know-how in the customer (a kind of intangible). The concept of `EconomicAgent` in REA needs not any extension in service transactions, and the distinction between "outside agents" (customers) and "inside agents" (employees) remains valid, as the accountability elements are equally valid (but not of specific relevance to model the service encounter). Nonetheless, the two specialized kinds of participations of agents are `provide` and `receive` and they relate an `EconomicAgent` to an `EconomicEvent`. This is somewhat limiting as it is not linking agents to resources in the context of events, i.e. it is not modeling ternary relationships. Such level of detail is needed to account for example for user-providing resources. For example, in consulting services, both the customer and the provider are usually providing some economic resources as valuable information or know-how, and there is a need to distinguish that different actors provide different resources to the same event. This should be accomplished in some way, and providing an extended service event structure (as described below) appears a good option to avoid changing the participation model in REA directly.

Resource Types

Even though the REA model seems to be valid for modeling resource exchange in services, the heterogeneity and special nature of resources in services deserve the elaboration of a detailed categorization of economic resources beyond the generic definition now available as "a thing that is scarce and has utility for economic agents and is something users of business applications want to plan, monitor and control". From the consideration of service as a kind of transfer described above, the next step is reviewing the different aspects that have been proposed in the literature to differentiate service from good transfer, and exercise REA modeling, eventually identifying *resource types* that are relevant enough in service to deserve separate representation. A particularly important subset of these resources is competencies, which can be defined as measurable capabilities required in performing some concrete work situations [10], but also physical elements that are not transferred but play a role in the service experience should be represented, e.g. facilities. Things as ontologies of competencies or even representations of social capital [4] could be used to characterize resources in a broad way, including intellectual capital. There are organizational theories that are relevant to resource classification and emphasize the role of some particular classes of resources given some contingent features as scarcity or non-substitutability. The Resource-based view of the firm (RBV) is a particularly interesting theory [3] that should be considered given that it explains organizational features based on the notion of resource. The current OWL version of the REA ontology is providing an `EconomicResourceType` concept accounting for a reification of types of resources that could be used as an alternative to define a terminology of resource types at the instance level. However, a detailed account of the typology of resources is needed for the REA to be more applicable in practice, avoiding the re-definition of kinds of resources that are recurring.

Kinds of Exchanges

The REA model considers a basic structure of events with two salient characteristics: events can be composite, and there is a relation of duality in the events, representing the two sides (producer-consumer) of the event. In REA, there are two types of interchanges: *transformations* and *transfers*. Transformations "create value through changes in form or substance" and "we either *use* or *consume* a resource to produce another", while transfers "create market value in a market transaction with outside parties". This is described in the model with the following sentence: "*A transfer fulfills a contract while a transformation fulfills a schedule*".

Transfers are basically targeted to model the transfer of ownership, which is closely connected with selling goods. Transformations are targeted to change, but they are mainly targeted to "back stage" transformations.

One possible option for modeling services is considering them a *kind of transfer*. In principle, there is nothing in the model that prevents us to do so:

1. The model does not preclude the provision of resources from the role of the customer, which is typical of many kinds of service encounters.
2. The model does not mandate that any physical good is necessarily part of the stock flow. What is transferred can be realized as an intangible, including elements aimed at producing some kind of satisfaction (e.g. in the case of a leisure service).

However, according to its definition, a transfer should model the transfer of value, so that for the models to be complete, there is a need to specify all the value creating steps, which are *ocurrents* bound in time [2], this is a requirement related to the inseparability of services from their actual provision. Figure 1 shows an example of a fragment of an interchange[7] where a typical REA dual purchase event is modeled with some added elements of service in a flight are also included[8] (in the dashed rectangle). It should be noted that these additional events are not in this case related to the sale event itself, but they are realizing the event. In the case of good transfer, the *use* stock flow is in many cases sufficient to represent the value transfer, but this is not the case in service. One option for this additional modeling is connecting outflow resources of type `ServiceCommitment` to the events realizing the service itself. Commitments in REA are defined as "a promise or obligation of economic agents to perform an economic event in the future". In consequence, a modification of transfers in REA can be done so that the buy of the service is modeled in the usual way with two events paired by a `duality` relation, but the outgoing stockflow is substituted by a `ServiceCommitment` that can be further described and that is described by the previous event.

However, this might be considered to be in contradiction with the necessity of pairing inflows and outflows stated by Geerts and McCarthy. Nonetheless, the alternative of having the commitment itself as outflow is not appropriate as commitments are not transferring by themselves the possession or temporal usage of resources. This separation of the event of buying the service from the service provision is convenient to model most service encounters, which are typically first agreed and later provided.

---

[7] It should be noted that the diagrams represent only fragments, abstracting many details for the sake of clarity.

[8] The notation used for the example is adapted from the original paper of Geerts and McCarthy (2000). Filled ovals are instances, and links between them are property instances with property names on top of them. Instance names and the concepts they belong are referenced as `instanceName:ConceptName`. This notation has been selected for ease of comparison with the diagrams in the previously mentioned REA paper.

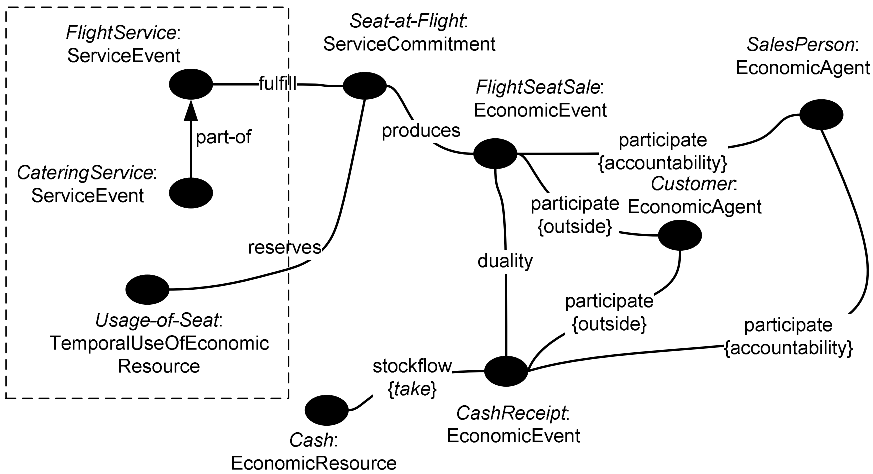


Fig. 1. An example REA representation of the sale of a flight seat

A `ServiceEvent` is a kind of `Event` but not an `EconomicEvent`, as it is not required to have a paired, dual event. However, the connection to a commitment allows tracing the original economic event to all the service events that are `fulfill`ing the commitment produced by the former. This separation and use of the commitment as an intermediate entity allows for a flexible modeling of service events structures. `ServiceEvents` can be found as a concept in OpenCyc, defined as "event in which one or more agents (related to the event via the predicate `providerOfService`) do something for one or more other agents (related to the event via the predicate `recipientOfService`". A relevant OpenCyc subclass is `ServiceProduct`, representing services done for payment. This definition captures the idea of "doing something for somebody" that need not necessarily be modeled together with the transfer of value itself (both can be considered two different levels of ontological concern).

Figure 1 is also showing how the commitment `reserves` some economic resource (in this case is a temporal usage of a resource, i.e. the seat, labeled as `Usage-of-Seat`). The use of intangibles in service and their eventual transfer requires some additional extensions. The actual flight service could be composed of some sub-services, using an standard `part-of` relationship. Following the above provide discussion on the output of services, the resources reserved can be used to infer the output flow of resources committed. This can be achieved in a straightforward way with SWRL[9] rules as the following:

```
ServiceCommitment(?sc) and DecrementEconomicEvent(?e)
and produces(?e, ?sc) and EconomicResource(?r)
and reserves(?sc, ?r)
-> outflow(?e, ?r)
```

[9] http://www.w3.org/Submission/SWRL/

However, as described above, this identification of resources used temporally as outflows can be considering conflicting with some interpretation of outflows in REA. In other direction, other rules can be used to define as outflows any resource used in the sub-events of the events fulfilling the commitment. This way, the outflows can be inferred from resource usage in the services, without a need to explicitly connecting them to the service commitment.

Figure 2 shows another example, in this case representing a coaching service. A relevant aspect in that model is that the *consultant* providing the service is both an instance of `EconomicAgent` (as it is participating in the service), but also a `EconomicResource` (actually, it is a temporal lending of the consultant to the service that constitutes the resource). This way, he/she has also links of participations with the main economic event, but it also participates in the service event themselves as a necessary resource.

Also in Figure 2, the *linkage* relationship is used to represent the key value-creating resources of the service providers (in the original model linkage is used to express relations between resources also), e.g. the know-how about the focus of the consultancy service. These resources are *required* in some of the events conforming the actual service provision. Further, these resources become outflows for the interchange of subtype `creates`, expressing that the know-how is not given (thus disappearing internally) but it is "copied" or reproduced (in this case, as the know-how is transferred to the customer). A common theme in service science is considering service as co-production of value, which in terms of the model entails that the customer role is also providing input flows to the transaction. This can be represented in a similar way with *linkage*s and requirements for some of the activities that are part of the service.

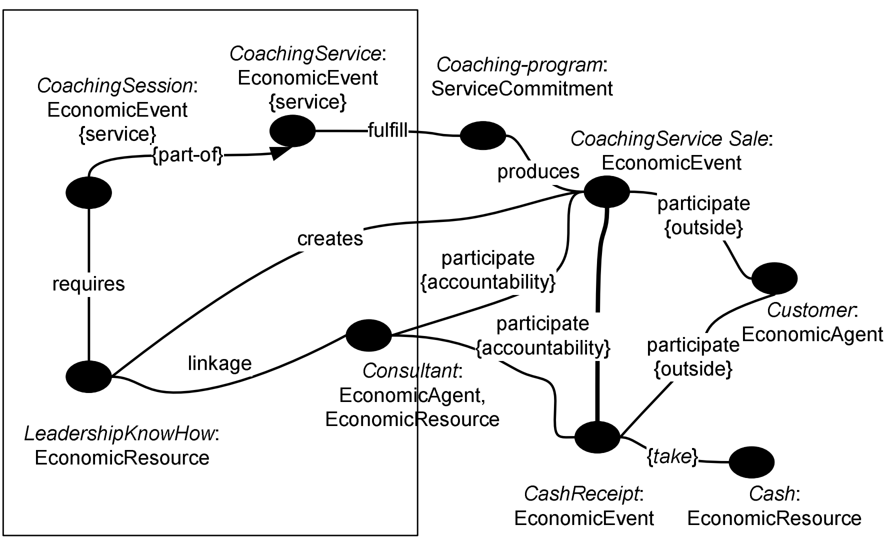


Fig. 2. An example REA representation of the sale and provision of a coaching service

Customer-provided resources are also typically part of service encounters. Following the example in Figure 2, a “business analysis” done by the customer as part of the process of consultancy could be considered a kind of `EconomicResource`. This example raises the concern if that resource could be associated to the inflow or the outflow of the service, as the analysis (the document) is done applying effort from the side of the customer, so it is providing something in, but it is done also with assistance of the consultant, so that it is not clearly an inflow or outflow. This example is a consequence of the intangibility of services, so that it is difficult to assess the direction of the flow of value in some cases. That “value directionality” paradox requires a much thorough analysis and a clarification of the notion of value transfer, beyond the duality of increment/decrement of value that has been introduced in the OWL version of the REA ontology.

4 Conclusions and Outlook

Services have been approached from different perspectives in previous research about service ontologies, emphasizing different aspects of services. In any case, service system ontologies require a sound modeling account of the interchange occurring in service encounters, which is the essential economic event addressed by design, management and measurement. The REA model addresses the dual structure of these events and the main elements necessary for their modeling, but requires some refinement to adequately support the conceptual structure of services in a explicit way. These refinements include the central notion of considering the temporal transfer of a resource to serve a specific customer as a kind of `outflow` that is not resulting in a transfer of ownership. This is consistent with opportunity costs as a consideration of resources, but it would require some changes in the ontology to represent the flow of available resources as services are being delivered. Other additions include a detailed model of the resources provided by different actors to the same event and the use of commitments to model the reserve of economic resources that are providing the service, separating in that way service buy from service provision. In general, the REA ontology appears as an appropriate point of departure for service systems ontology, as it is representing value transfer appropriately and has the potential of accommodating the specifics of services, but cannot be reused “as is” as there are several conceptual issues that remain unclear.

The analysis provided here is provisional, and it would require extensive evaluation by contrasting service theories and models, and by developing complex service provision case studies, that would be subject to future work. In any case, the analysis provides a point of departure for reusing the REA model in the SSME domain by correcting, extending or modifying the main aspects described above.

Further work is needed in developing a complete extension of the REA enterprise ontology for services, or perhaps developing some guidelines and links to other ontologies that complement what the REA is offering as an abstract model for value exchange. Also, future work should address how software services can be integrated with the account of services provided here.

References

1. Alter, S.: Service system fundamentals: Work system, value chain and life cycle. IBM Systems Journal 47(1), 71–85 (2008)
2. Arp, R., Smith, B.: Function, role, and disposition in basic formal ontology. In: Proceedings of the 11th Annual BioOntologies Meeting (2008)
3. Barney, J.B.: Firm resources and sustained competitive advantage. Journal of Management 17, 99–120 (1991)
4. Burt, R.: The Social Capital of Structural Holes. In: Guillén, M.F. (ed.) The New Economic Sociology: Developments in an Emerging Field, pp. 148–190. Russell Sage Foundation, New York (2002)
5. Buxmann, P., Hess, T., Lehmann, S.: Software as a Service. Wirtschaftsinfor-Matik 50(6), 500–503 (2008)
6. Chandrasekaran, B., Josephson, J., Benjamins, R.: What are ontologies, and why do we need them. IEEE Intelligent Systems 14(1), 20–26 (1999)
7. Chen, J.S., Tsou, H.T.: Information technology adoption for service innovation practices and competitive advantage: the case of financial firms. Information Research 12(3), Paper 314 (2007)
8. Chesbrough, H., Spohrer, J.: A research manifesto for services science. Communications of the ACM 49(7), 35–40 (2006)
9. Da Silveira, G., Borenstein, D., Fogliatto, F.S.: Mass customization: Literature review and research directions. International Journal of Production Economics 72(1), 1–13 (2001)
10. Dzinkowski, R.: The measurement and management of intellectual capital: An introduction. Management Accounting 78(2), 32–36 (2000)
11. Feigenbaum, J., Parkes, D.C., Pennock, D.M.: Computational challenges in e-commerce. Communications of the ACM 52(1), 70–74 (2009)
12. Ferrario, R., Guarino, N.: Towards an Ontological Foundation for Services Science. In: Domingue, J., Fensel, D., Traverso, P. (eds.) FIS 2008. LNCS, vol. 5468, pp. 152–169. Springer, Heidelberg (2009)
13. Ferrario, R., Guarino, N., Fernández Barrera, M.E.: Towards an Ontological Foundation for Services Science: the Legal Perspective. In: Sartor, G., Casanovas, P., Biasiotti, M., Fernandez Barrera, M. (eds.) Approaches to Legal Ontologies, Springer, Heidelberg (2009)
14. Geerts, G., McCarthy, W.E.: An Accounting Object Infrastructure For Knowledge-Based Enterprise Models. IEEE Intelligent Systems & Their Applications (July/August 1999), 89–94 (2000)
15. Geerts, G.L., McCarthy, W.E.: The Ontological Foundation of REA Enterprise Information Systems. Paper presented at the American Accounting Association Conference Philadelphia, PA (2000), https://www.msu.edu/user/mccarth4/Alabama.doc [retrieved online November 15, 2009]
16. Gruber, T.: The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In: Allen, J.A., Fikes, R., Sandewall, E. (eds.) Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, pp. 601–602. Morgan Kaufmann, Cambridge (1991)
17. Hepp, M.: Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. International Journal on Semantic Web & Information Systems 2(1), 72–99 (2006)
18. Lenat, D.: Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)

19. McCarthy, W.: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review, 554–578 (1982)
20. von Mises, L.: Human Action: A Treatise on Economics, 3rd edn. Henry Regnery, Chicago (1966)
21. Mora, M., Raisinghani, M., Gelman, O., Sicilia, M.A.: Onto-ServSys: a Service System Ontology. In: Demirkan, H., Spohrer, J., Krishna, V. (eds.) The Science of Service Systems/Service Systems Implementation. Service Science: Research and Innova-tions (SSRI) Series. Springer, Heidelberg (to appear)
22. Teboul, J.: Service is Front Stage: Positioning Services for Value Advantage. Insead Business Press, Palgrave Macmillan (2006)

Author Index