Marcello P. Bax, Maurício B. Almeida, Renata Wassermann (Eds.)

VI Seminar on Ontology Research in Brazil

Belo Horizonte, Brazil, September 23-25, 2013
Proceedings

http://ontobras.eci.ufmg.br/en/

Sponsors:

Organizing Institutions:

Supporters:

Editors' addresses:

Marcello P. Bax — bax@eci.ufmg.br
Escola de Ciência da Informação — Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 — Campus Pampulha
CEP 31270-901— Belo Horizonte, MG — Brazil

Maurício B. Almeida — mba@eci.ufmg.br
Escola de Ciência da Informação — Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 — Campus Pampulha
CEP 31270-901— Belo Horizonte, MG — Brazil

Renata Wassermann — renata@ime.usp.br
Instituto de Matemática e Estatística — Universidade de São Paulo (USP)
Rua do Matão, 1010 — Cidade Universitária
CEP 05508-090 — São Paulo, SP — Brazil

Preface

Ontology is a cross-disciplinary field concerning the study of concepts and theories that support the building of shared conceptualizations of specific domains. In recent years, there has been a growing interest in the application of ontologies to solve modeling and classification problems in diverse areas such as Computer Science, Information Science, Philosophy, Artificial Intelligence, Linguistic, Knowledge Management and many others.

The Seminar on Ontology Research in Brazil, ONTOBRAS, foresees an opportunity and scientific environment in which researchers and practitioners from Information Sciences and Computer Science can discuss the theories, methodologies, languages, tools and experiences related to ontologies development and application.

Particularly, this Seminar Six Edition took place in an Information Science school, as the result of an effort of the ontologies research community in exchanging experiences and in integrating Information Science and Computer Science initiatives.

The event was organized by the *Federal University of Minas Gerais* (UFMG), *Information Science School* (ECI). It was also supported by *National Association of Research and Graduate Programs in Information Science* (ANCIB) and by *Fumec University*. The event was partially funded by *CAPES Foundation* and by the *National Council of Research* (CNPQ) both from the Brazilian Education Ministry, and by *Foundation for the Support of Research of the State of Minas Gerais* (FAPEMIG).

Researchers and practitioners were invited to submit theoretical, technical and practical research contributions that directly or indirectly address the issues above. The call for papers was open for two categories of submissions: Full papers (maximum 12 pages) written in English and describing original work with clear demonstrated results. Accepted full paper were invited for oral presentation. The second category was short papers (maximum 6 pages), written in Portuguese, or English, or Spanish and describing ongoing work. Accepted short papers were be invited for poster presentations.

We received 52 submissions, out of which 13 were accepted for publication and oral presentation; and 10 were accepted for publication and poster presentations. This volume is thus constituted by13 full papers and 10 short papers, selected by our program committee, which is composed by national and international referees.

We thank the organizing committee for their commitment to the success of the event, the authors for their submissions and the program committee for their hard work.

September 2013

Marcello P. Bax
Renata Wassermann
Mauricio B. Almeida

General Chairs

Marcello Peixoto Bax (ECI - UFMG)
Fernando Silva Parreiras (FUMEC)

Program Chairs / Editorial Chairs

Maurício Almeida (ECI - UFMG)
Renata Wassermann (IME - USP)

Organizing Committee

Andreia Malucelli (PUCPR, Brazil)
Bernadette Loscio (UFPE, Brazil)
Fernando Gauthier (UFSC, Brazil)
Fred Freitas (UFPE, Brazil)
Giancarlo Guizzardi (UFES, Brazil)
Mara Abel (UFRGS, Brazil)
Marcello Bax (UFMG, Brazil)
Maria Claudia Cavalcanti (IME, Brazil)
Maria Luiza de Almeida Campos (UFF, Brazil)
Renata Vieira (PUCRS, Brazil)
Sonia Elisa Caregnato (UFRGS, Brazil)

Local Organizing Committee

Luiz Gustavo Fonseca Ferreira (ECI - UFMG)
Renata Baracho (ECI - UFMG)
Eliza Tuler (ECI - UFMG)
Cátia Rodrigues (ECI - UFMG)
Benildes Maculan (ECI - UFMG)

Program Committee

Mauricio Almeida (UFMG, Brazil)
Alan Pedro da Silva (UFAL, Brazil)
Fernanda Araujo Baiao (UNIRIO, Brazil)
Marcello P. Bax (UFMG, Brazil)
Ig Ibert Bittencourt (UFAL, Brazil)
Stefano Borgo (Laboratory for Applied Ontology, Italy)
Regina Braga (UFJF, Brazil)
Marisa Brascher (UnB, Brazil)
Ligia Café (UFSC, Brazil)
Gilberto Camara (INPE, Brazil)
Maria Luiza De Almeida Campos (UFF, Brazil)

Sônia Caregnato (UFRGS, Brazil)
Rove Chishman (Unisinos, Brazil)
Oscar Corcho (UPM, Spain)
Evandro Costa (UFAL, Brazil)
Paulo Costa (George Mason University, United States)
Alicia Diaz (LIFIA, Argentina)
Jérôme Euzenat (INRIA, Grenoble, Ródano-Alpes, France)
Ricardo Falbo (UFES, Brazil)
Roberta Ferrario (Laboratory for Applied Ontology, Italy)
Frederico Fonseca (Penn State, United States)
Fred Freitas (UFPE, Brazil)
Renata Galante (UFRGS, Brazil)
José Augusto Chaves Guimarães (UNESP, Brazil)
Claudio Gutierrez (University of Chile, Chile)
Gabriela Henning (INTEC/UNL, Argentina)
Wolfgang Hesse (Philipps - University Marburg, Germany)
Seiji Isotani (University of São Paulo, Brazil)
Nair Kobashi (USP, Brazil)
Werner Kuhn (University of Muenster, Germany)
Fernanda Lima (UnB, Brazil)
Gercina A B O Lima (UFMG, Brazil)
Vânia Lima (USP, Brazil)
Maria Machado (UFRJ, Brazil)
Andreia Malucelli (PUC-PR, Brazil)
Riichiro Mizoguchi (Osaka University, Japan)
Regina Motz (Universidade de la Republica, Uruguai)
Ana Maria Moura (LNCC, Brazil)
Fernando Naufel (UFF, Brazil)
Alcione Oliveira (UFV, Brazil)
José Palazzo M. De Oliveira (UFRGS, Brazil)
Jose M. Parente De Oliveira (ITA, Brazil)
Fernando Parreiras (University of Koblenz-Landau, Germany)
Paulo Pinheiro Da Silva (University of Texas at El Paso, United States)
Fabio Porto (LNCC, Brazil)
Florian Probst (SAP, Germany)
Renato Rocha Souza (FGV-RJ, Brazil)
Ana Carolina Salgado (UFPE, Brazil)
Plácida da Costa Santos (UNESP, Brazil)
Stefan Schulz (University of Graz, Austria)
Daniel Schwabe (PUC-RJ, Brazil)
Renata Vieira (PUCRS, Brazil)
Gerd Wagner (Brandenburg University of Technology at Cottbus,Germany)
Renata Wassermann (USP, Brazil)

Contents

Part I

Full Papers

Hemocomponents and Hemoderivatives Ontology (HEMONTO): an Ontology About Blood Components

Fabrício M. Mendonça[1], Maurício B. Almeida[1]

[1]Escola de Ciência da Informação – Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 - Campus Pampulha – 31.270-901 – Belo Horizonte – Brazil

fabriciommendonca@gmail.com, mba@eci.ufmg.br

Abstract. *Ontologies has been widely used in the formal description of scientific knowledge, as well in the practice of the conceptual modeling. Considering the description of scientific knowledge, several ontologies has been proposed in the biomedical domain. This article describes an ongoing research in the domain of blood transfusion, presenting the construction of a domain ontology about hemocomponents and hemoderivatives. Such ontology, named HEMONTO, has been developed using top-level ontologies, biomedical domain ontologies, among other resources. The ontology is based in a set of philosophical principles that has been identified in the literature under the label "ontological realism"and relies on technologies developed in the scope of Semantic Web. HEMONTO aims to provide both a knowledge repository about blood transfusion and an auxiliary instrument for modeling and evaluation of the information systems. The results presented here refer to the partial content of the ontology, encompassing classes, relations and representation diagrams.*

1. Introduction

The search for the best way to represent reality in information systems has been constrained over the years by the intrinsic limitations of the modeling techniques. The inconsistency of the modeling activity during the first years of the conceptual modeling, may have been the reason for many of the current interoperability problems between systems [Smith and Welty 2001]. This situation becomes more complex when one can see that the practices of information systems conceptual modeling has been oriented to specific modeling cases and performed in an ad-hoc way[Fonseca and Martin 2007].

Ontologies has been proposed as an alternative to relieve this type of problem. Indeed, the use of ontologies represents an evolution in the practices of information systems modeling of [Guarino 1998] [Smith 2003] [Wand and Weber 2004] [Fonseca and Martin 2007]. Ontologies allow to make explicit the acquired knowledge from a domain, promoting the sharing of knowledge and supporting the integration of information between different representation instruments, such as information systems.

This article describes an ongoing research in the domain of hematology and blood transfusion. The research encompasses a case study, which purpose is the construction of an ontology about human blood components (hemocomponents and hemoderivatives), named HEMONTO. We expect that the outcome of our research may be used as a scientific knowledge repository, for example as an annotation instrument [Rubin et al. 2008] and, accordingly, to fill a lack caused by missing of the formal representation geared for blood components and by limited possibilities of information

retrieval in this area as function of the use of general tools. Moreover, we expect that the ontology can facilitate the modeling activities or the information systems evaluation in the blood transfusion domain. This paper presents the partial content of HEMONTO – set of terms, representation diagrams, some semi-formal definitions – which has been developed in the scope of the Blood Project [ALMEIDA et al., 2010]. It is also worth mentioning the project guidelines, which have been used in the development of HEMONTO: it can be categorized as an ontology for information systems [Fonseca 2007] or as an ontology-driven information systems [Guarino 1998]; it relies on philosophical principles, which has sometimes received, in th literature that deals with biomedical ontologies, the label "ontological realism" [Smith and Ceusters, 2010].

The remaining part of the article is organized as follows: section 2 presents the required research background, explaining the basis of domain and high-level ontologies used here; section 3 describes the methodology used for the development of the ontology; the section 4 presents the partial content of HEMONTO, developed so far; and, finally, section 5 presents a brief discussion about the study and possibilities for future works.

2. Background

Ontology is a topic that has been studied for a long time in Philosophy, where it is defined as branch of metaphysics that deal with things existing in world. In Computer Science, ontologies are considered a software engineer artifact. In Information Science, ontology is seen as a type of controlled vocabulary used for information retrieval. In the literature under the topic "ontology", it be found many definitions for the term in different publications, such [Grüber 1993], [Guarino 1998], [Soergel 1997], [Vickery 1997], [Sowa 2000], to mention but a few. The approaches range from the philosophical bias to the context of the information systems.

One of the principles universally accepted in the construction of ontologies is the reuse of terms and relations from other ontologies. In general, a domain ontology is developed using both the hierarchy of a high-level ontology and terms obtainned from other domain ontologies. In the development of HEMONTO, we chose the following ontologies: (i) *Basic Formal Ontology* (BFO); (ii) *Relation Ontology* (RO) e (iii) *Foundational Model of Anatomy* (FMA). These choices wer based in the fact that there are a large number of biomedical ontologies grounded on those generic ontologies[1].

The Basic Formal Ontology (BFO) received influence from Aristoteles' works and from Edmund Husserl's metaphysics and logics. Other ontologies also contributed in the creation of BFO, like the DOLCE [Masolo et al., 2003]; Also, BFO has a philosophical ground based on the so-called ontological realism [Grenon and Smith 2004]. Hence, the desired interpretation for the BFO fundamental entities and relations is that they are real divisions among types of entities existing in the world. In addition, these entities are independent of the human mind [Spear 2006]. Accordingly, categories in the BFO are entities termed universals. The BFO also includes particulars, that is, the instances of those universals.

[1] See, for example, available initiatives in the *OBO Foundry* (http://www.obofoundry.org/) and in the *BioPortal* (http://bioportal.bioontology.org/)

The BFO universals are grouped in two different branches: (i) *continuants:* entities that endure through time while maintaining their identity and that have no temporal parts (examples: a human individual, the human blood, the disposition of an organism to bleed); and (ii) *occurrents*: entities that happen, unfold or develop in time and that have temporal parts (examples: the process of respiration, a whole human life in the 19th century, the functioning of a heart) [Grenon and Smith 2004] [Spear 2006]. The BFO entities are linked together by the ontological relations defined in the RO, which were incorporated in the BFO semantic structure. Indeed, the ontological relations of the RO and BFO are the same.

The RO is result of collaborative work accomplished by groups of research in biomedical ontologies (BFO, Gene Ontology[2], FMA e GALEN), with the purpose of defining a restrict set of relations to be used in biomedical ontologies. These relations are logically well-defined and created in order to fostering interoperability [Smith and Ceusters 2010]. In its first version, in 2005, the RO was published containing 10 formal relations in the biomedical domain and. In the current version[3], it has a total of 160 relations.

In the construction of HEMONTO, we make use of the ontological relations present in RO. These relations establish the basic connections among classes (<class, class>), among instances (<instance, instance>) and among classes and instances (<instance, class>) [Smith et al. 2005]. The term *class* will be used, henceforward, to refer to an entity in the reality equivalent to the terms *universal* and *type*, considering that the main ontology editors do not make this distinction. Similarly, the term *instance* will be used to refer to a particular in reality, which is equivalent to the terms *particular* and *individual*.

Other ontology that is important in our work is the *Foundational Model of Anatomy (FMA)*. It was created as a set of classes necessary for the symbolic representation of phenotypic structure of the human body, specifically, the anatomy. The FMA was developed based both on some fundamental modeling principles (unified context, abstraction level, definition principle, dominant concept) and on aristotelian definitions about the objects of the world [Rosse and Mejino 2003]. As a consequence of this approach, the nodes of the FMA hierarchies are called of *classes* or *types*, bolstering its commitment with entities of the real world, instead of commitment with the meanings of the terms. Currently, the FMA contains about 75.000 classes, which represent entitities like complex macromolecular structures, cell components of the human body, and so forth; about 120.000 terms associated with these classes and 168 types of relationships [FMA 2013].

The content related to "blood" in the FMA includes entities as blood itself (*FMA: blood*) and some of its specifications, such as *FMA: Venous blood* and *FMA: Plasma*. Despite the FMA includes some entities of the blood domain, the convering of this domain is shallow, not including specific components such as hemocomponents and hemoderivatives. Within our study, some FMA terms are used in the ontology HEMONTO as a starting point for the definition of more general terms.

[2] Available in: http://www.geneontology.org/. Access: 07th of May 2013.
[3] Available in: http://code.google.com/p/obo-relations/. Access: 26th of April 2013.

3. Methodology

In this section, we describe the methodological steps performed to conduct this research. Basically, these steps are: (i) a study in the domain of the human blood, using bibliographic resources and reference publications in the hematology field; (ii) a literature review, followed by an exploratory study of the relevant ontologies in the biomedical domain; and (iii) the construction of the ontology *per se*, using other ontologies and principles of top-level ontologies.

For the study of the hematology domain, we selected initially the "ISBT 128 - Standard Terminology for Blood, Cellular Therapy, and Tissue Product Descriptions" [ICCBBA 2010], since it is the reference manual about blood and cellular therapy. This document provides a standard terminology for describing transfusion and transplantation products. It is designed to allow distinction between products where such is required on safety, clinical practice, or inventory management grounds.

Parallel to the study of the blood domain, we performed a literature review about relevant ontologies in the domain of blood. Thus, we selected those ones more suitable for our approach, which includes BFO, RO e FMA. Specifically, the criteria used in the selection of these ontologies were: (i) its scope of coverage; (ii) compatibility with ontological realism, which is adopted in the project; (iii) current applicability of these ontologies; (iv) available content (classes, relations and axioms freely accessable); and (v) underlying principles and logic formalisms.

With regard to the use of classes and relations of other ontologies, we proceeded as follows: (i) the BFO was used for the definition of generic classes; (ii) the relations of the RO were used the basis for the composition of the HEMONTO relations; and (iii) the classes of the FMA were used in the definition of specific classes in the blood domain. For the knowledge representation of the in the blood domain, we constructed taxonomies (using formal relation **is_a**), partonomies (using formal relation **part_of**) and other relevant ontological relations, such as **participates_in, has_agent, produces, has_quality**. The ontology's editor Protege 4.2[4] was used for the construction of the ontology and it enabled the implementation of the ontology in *Ontology Web Language* (OWL). For the creation of the diagrams (taxonomies, partonomies and others) of the ontology proposed, we used the software Diagram Editor[5].

It is worth mentioning the use of a semi-formal syntax for specify the ontological relations, according to the guidelines suggested by ontologies integrated to the repository *Open Biomedical Ontologies (OBO)*. This syntax involves a set of basic conventions of the logic notation described in Smith et al. (2005). Here, this logic notation was used with some adaptations. The basic conventions are:

- The variables C, C1, ... Cn (capital letter) are used for the representation of *continuants universals* and c, c1, ... cn (lower-case letter) are used for the representation of *continuants particulars*;
- The variables P, P1, ... Pn (capital letter) are used for the representation of *occurrents universals* and c, c1, ... cn (lower-case letter) are used to the representation of *occurrents particulars*;

[4] Available in: http://protege.stanford.edu/. Access: 03rd of September 2013.
[5] Available in: https://projects.gnome.org/dia/. Access: 03rd of September 2013.

- The variables t, t1, ... tn are used to represent intervals.
- The relations between two universals (for example: C **is_a** C1), between two particulars (for example: c1 **part_of** c) and between one particular and one universal (for example: c **instance_of** C) are all in bold.

This notational pattern adopted here is especially important in the definition and understanding of the RO ontological relations, used to link terms. Altogether, we used 9 ontological relations to link HEMONTO terms, which were defined according to Smith et al. (2005) as: (see Table 1)

Table 1. Semiformal definitions for the ontological relations included in HEMONTO.

Relation	Semiformal Definition	Examples HEMONTO
C **is_a** C1	∀c, ∀t, if c **instance_of** C at t then c **instance_of** C1 at t, where c **instance_of** C at t is a primitive relation, at which a continuant particular c instance the universal C in the given time t.	Hemocomponent **is_a** Object Aggregate. Platelet concentrate **is_a** Hemocomponent.
P **is_a** P1	∀p, if p **instance_of** P then p **instance_of** P1, where p **instance_of** P is a primitive relation, at which a occurrent particular p instance the universal P.	Process of freeze **is_a** Process. Process of centrifugation at high rotation **is_a** Process of centrifugation.
C **part_of** C1	**∀**c, **∀**t, if c **instance_of** C at t then there is some c1 such that c1 instance of C1 at t and c **part_of** c1 at t, where c **part_of** c1 at t is a primitive relation between two continuant instances and a time at which the one is part of the other.	Plasma **part_of** Whole portion of blood. Fibrinogen **part_of** Cryoprecipitate
C **participates_in** P	∀c, ∀t, if c **instance_of** C at t then there is some p such that p **instance_of** P and p **has_participant** c at t, where p **has_participant** c at t is a primitive relation between a process, a continuant, and a time at which the continuant participates in some way in the process.	Whole portion of blood **participates_in** Process of centrifugation. Plasma **participates_in** Cryoprecipitate extraction.
P **produces** C	**∀**p, if p **instance_of** P then there is some c, t; such that if c **instance_of** C1 at t and p **produces** c at t, where p **produces** c at t is a relation between a process p, a continuant c and a time t, at which p produces c if some process that **occurs_in** p **has_output** c.	Process of centrifugation **produces** Buffy coat. Cryoprecipitate extraction **produces** Cryoprecipitate free plasma.
P **preceded_by** P1	**∀**p, if p **instance_of** P then there is some p1 such that p1 **instance** of P1 and p **preceded_by** p1, where p **preceded_by** p1 = **∀**t, **∀**t1, if p **occurring_at** t and p1 occurring_at t1, then t1 earlier t, where t **earlier** t1 is a primitive relation between two times such that t occurs before of t1 and **p occurring_at t** = for some c, p **has_participant** c at t.	Process of centrifugation at high rotation **preceded_by** Process of centrifugation. Process of collection **preceded_by** Process of centrifugation.
C **contained_in** C1	**∀**c, **∀**t, if c **instance_of** C at t then there is some c1 such that: if c1 **instance_of** C1 at t1 and c **contained_in** c1 at t, where c **contained_in** c1 at t = c **located_in** c1 at t and not c **overlap** c1 at t	Plasma **contained_in** Top and bottom pocket. Erythrocyte **contained_in** Top and bottom pocket.
P **has_agent** C	**∀**p, if p **instance_of** P then there is some c, t; such that if c **instance_of** C1 at t and p **has_agent** c at t, where p **has_agent** c at t is a primitive relation between a process, a continuant and a time at which the continuant is causally active in the process.	Extraction of buffy coat **has_agent** Plasma extractor.
C **has_quality** Q	Relation between an continuant entity C and a quality Q, at which C **has_quality** Q if only if: **∀**c, **∀**t, if c **instance_of** C at t then there is some c1 such that: if c1 **instance_of** C1 and exists **∀**q, **∀**t, if q **instance_of** Q at t then there is some q1 such that: if q1 **instance_of** Q1, such that q	Fresh frozen plasm **has_quality** Time after collection. Plasma of 24 hours **has_quality** Freeze time.

	inheres_in c at t.	
Q **is_quality_measured_as** q	Relation between a continuant universal Q and a continuant particular q, such that both are qualities, and ∀q, ∀t, if q **instance_of** Q at t then there is some q1 such that q1 **instance_of** Q1.	Time after collection **is_quality_measured_as** <= 8 hs. Freeze time **is_quality_measured_as** > 1 h.

4. Results

In this section, we present the partial results obtained so far. We describe the ontology about hemocomponents and hemoderivatives of the human blood, highlighting classes, relations and diagrams used to represent the knowledge. As we mentioned, these results are partials, since the process of construction of ontologies, especially in complex domains, should not be considered finished and must be under constant evaluation [Grüber 1993].

HEMONTO represents knowledge about blood products – hemocomponents and hemoderivatives – encompassing the constituent elements those products, as well as the procedures used to obtain them. In its current version, the ontology has 54 terms, which 45 are classes of the ontology and other 9 are relations. Among the classes, 30 classes are specific of the ontology, 13 classes were imported from the BFO and 2 classes from the FMA. The relations were imported from the RO and adapted to the domain under study.

In the remainder of this section, we present the classes and relations of HEMONTO, as well as the representation structures connecting them. Each ontology class or relation, when referenced in the text, is represented in italic. Similarly, classes and relations imported from other ontologies are spelled in italic and accompanied of an acronym representing the source ontology.

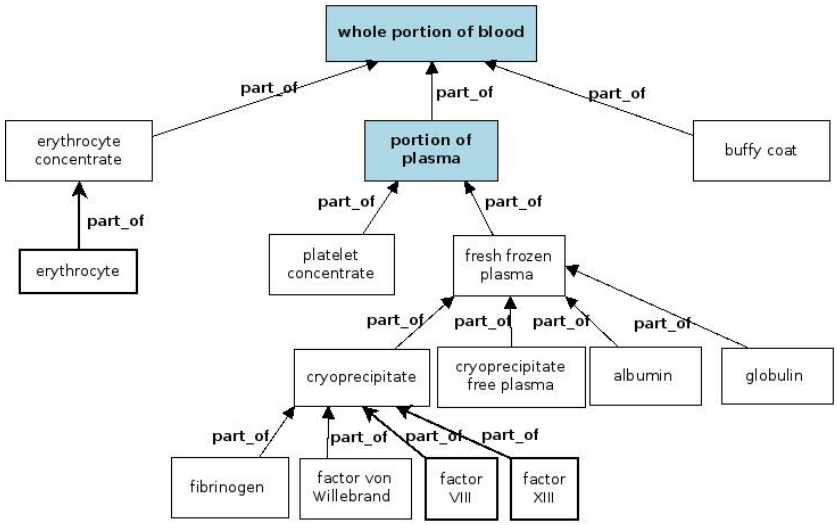


Figure 1. Partonomy of the blood components.

According to the FMA, the *blood (FMA: portion of blood)* is the substance and main fluid of the human body, composed of plasma (*FMA: portion of plasma*) and blood cells. In order to obtain hemocomponents and hemoderivatives is necessary to submit one unity of the *whole blood (FMA: whole portion of blood)* to specific processes, such as centrifugation and freeze. According to the FMA, the *whole portion of blood* is a type of *portion of blood*, such as its components were not separated.

Aiming to understand these processes and the types of hemocomponents and hemoderivatives that can be obtained, we constructed a partonomy of the blood derivate products (see figure 1). In this partonomy, the entities originated from the FMA are represented by shaded rectangles and the other entities, specific of HEMONTO, are represented as rectangles without shading.

The partonomy demonstrates that the *whole portion of blood*, when subjected to the first process of centrifugation, is separated, initially, in three products: (i) *erythrocyte*, whose a portion, when stored in specific conditions of temperature and storage, generates the hemocomponent *erythrocyte concentrate*; (ii) *portion of plasma*, which corresponds to *plasma* in its natural state (gross) yet rich in platelets (synonym term: *platelet rich plasma*); and (iii) *buffy coat*, a portion of blood formed by leucocytes and platelets. Next, after a new centrifugation process applied to a portion of blood performed in high rotation, the *portion of plasma* is separated into two products: (i) the *platelet concentrate* and (ii) the *fresh frozen plasma* (a plasma with low percentage of platelets). On the other hand, the *fresh frozen plasma* can be submit to a extraction process of one of its own components – the *cryoprecipitate* – creating two other hemocomponents: (i) the *cryoprecipitate* and (ii) the *cryoprecipitate-free plasm*. From *fresh frozen plasma,* it is still possible to extract two hemoderivatives – *albumin* and *globulin* – from the plasma subdivision by industrial process. Finally, the hemocomponent *cryoprecipitate* has glycoproteins of high molecular weight (*fibrinogen, factor Von Willdebrand, factor VIII* and *factor XIII*) that fulfill the role of clotting factors in the blood transfusion process. Using a industrial process it is possible both to obtain these proteins and to generate other important hemoderivative named *clotting factors concentrate*, which encompasses these proteins.

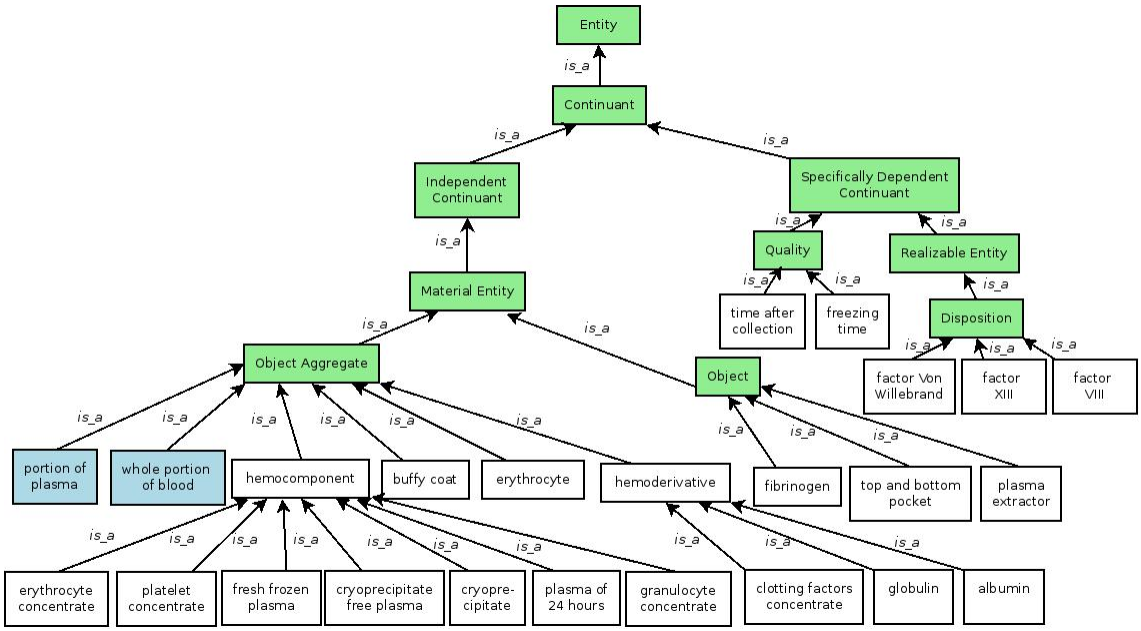


Figure 2: Taxonomy of the continuants entities of HEMONTO.

According to the taxonomic structure of the BFO, we have two large groups of real entities: the *continuants* and the *occurrents*. Following this structure, we created two taxonomies of entities included in HEMONTO, taking as a starting point the fundamental categories of the BFO. The taxonomy of figure 2 represents the set of

continuant entities of HEMONTO, in which the entities imported from BFO and FMA are represented as shaded rectangles and the other entities, specific of HEMONTO, are represented as rectangles without shading.

According to the BFO classification, the entity *FMA:whole portion of blood is_a BFO:object aggregate*, similarly to its components (*FMA:portion of plasma, buffy coat* and *erythrocyte*) obtained in the first process of blood centrifugation. The *hemocomponents* and the *hemoderivatives* of the human blood also are classified as *BFO:object aggregate*, as well as its specific types: (i) *erythrocyte concentrate, platelet concentrate; fresh frozen plasm; cryoprecipitate free plasma, cryoprecipitate, plasma of 24 hours* and *granulocyte concentrate*, which are types of *hemocomponents*; and (ii) *clotting factors concentrate, globulin* and *albumin*, which are types of *hemoderivatives*.

The entities *plasma extractor* and *top and bottom pocket*, used in the production of the hemocomponent *platelet concentrate*, are classified as *BFO:object*, similarly to protein *fibrinogen*, contained in the *cryoprecipitate*. This blood component still contains elements as *factor Von Willebrand, factor VIII* and *factor XIII*, which work as clotting factors and therefore were classified as *BFO:disposition*. The entities *time after collection* and *freezing time* were classified as *BFO:quality*, since these entities are important parameters of the hemocomponents during their process of production.

The other large group of entities of HEMONTO corresponds to *occurrent entities*. This group is represented in the taxonomy depicted in figure 3:

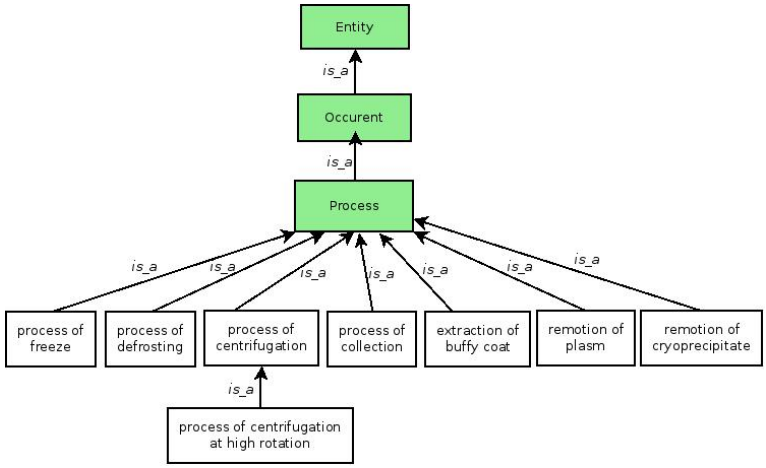


Figure 3: Taxonomy of the occurrents entities of HEMONTO.

Within the taxonomy of *occurrent entities* (figure 3), again, the entities extracted from BFO are represented as shaded rectangles and the entities specific from HEMONTO are represented as rectangles without shading. In this taxonomy, we tried to include all process involved in the production of human blood hemocomponents and hemoderivatives: *process of freeze; process of defrosting; process of centrifugation* and specific types as *process of centrifugation at high rotation*; *process of collection*; *extraction of buffy coat; remotion of plasma* and *remotion of cryoprecipitate*. All these entities were classified as *BFO:process*, since their existence are connected to an event or an occurrence. In addtion, they have their own temporal parts and dependence of one or more material entities, according to Grenon and Smith (2004).

In addition to the partonomy and the taxonomy presented, we needed to create other representation structures involving different ontological relations as a way to

describe the specificities involved in the processes of production of each blood hemocomponents and hemoderivatives. The processes of production of the hemocomponents *erythrocyte concentrate* and *platelet concentrate* are represented in the diagram of figure 4:

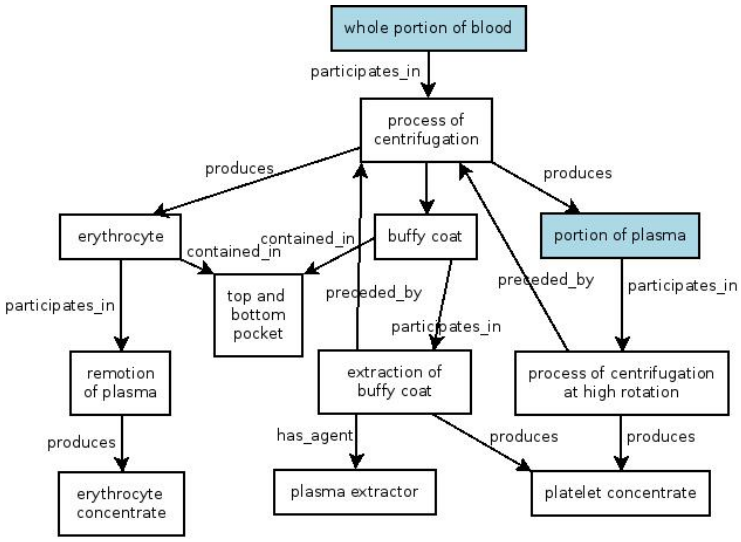


Figure 4: Processes for obtaining of the erythrocytes and platelet concentrate.

The initial procedure to obtain both hemocomponents (figure 4) consists in the *process of centrifugation* of the *whole portion of blood,* which separates the following elements: *portion of plasma, buffy coat* and *erythrocyte*. Therefore, we represented that the *whole portion of blood* **participates_in** the *process of centrifugation,* which, on the other hand, **produces** *portion of plasma, buffy coat* and *erythrocyte*. In order to obtain the *erythrocyte concentrate* (left side of figure 4), plasma, **contained_in** *top and bottom pockets*, is removed of the set of *erythrocytes* that remained after the *process of centrifugation* of the *whole portion of blood*. Thus, we represented the *remotion of plasma* **produces** *erythrocyte concentrate*.

However, in order to obtain the hemocomponent *platelet concentrate* (right side of figure 4), two different methods can be used: (i) obtaining it from *buffy coat*; and (ii) obtaining it from *plasma*. In the first method, the *buffy coat* **contained_in** *top and bottom pockets* after the *process of centrifugation*, is extracted by one the outputs of the *top and bottom pocket* with the use of *plasm extractors*. Therefore, it was represented that the *extraction of buffy coat* **has_agent** *plasm extractors* and that the *extraction of buffy coat* **produces** *platelet concentrate*. In the second method, the *plasma* obtained after the *process of centrifugation* (called light centrifugation) is again centrifuged in high rotation (*process of centrifugation at high rotation*). After this process, it **produces** *platelet concentrate*.

The plasma is one of the most important components of the human blood and, as result of process acomplished on it, one can generates other four blood hemocomponents – (i) the *fresh frozen plasm*, (ii) the *cryoprecipitate free plasm*, (iii) the *plasma of 24 hour*, (iv) the *cryoprecipitate* – and also three blood hemoderivatives – (i) the *albumin*; (ii) the *globulin* and (iii) the *clotting factors concentrate*. The diagram depicted in figure 5 represents the processes required for the achievement of the mentioned hemocomponents and hemoderivatives. The rectangles of the figure 5

represents classes of the ontology and the ellipses represents the properties of these classes.

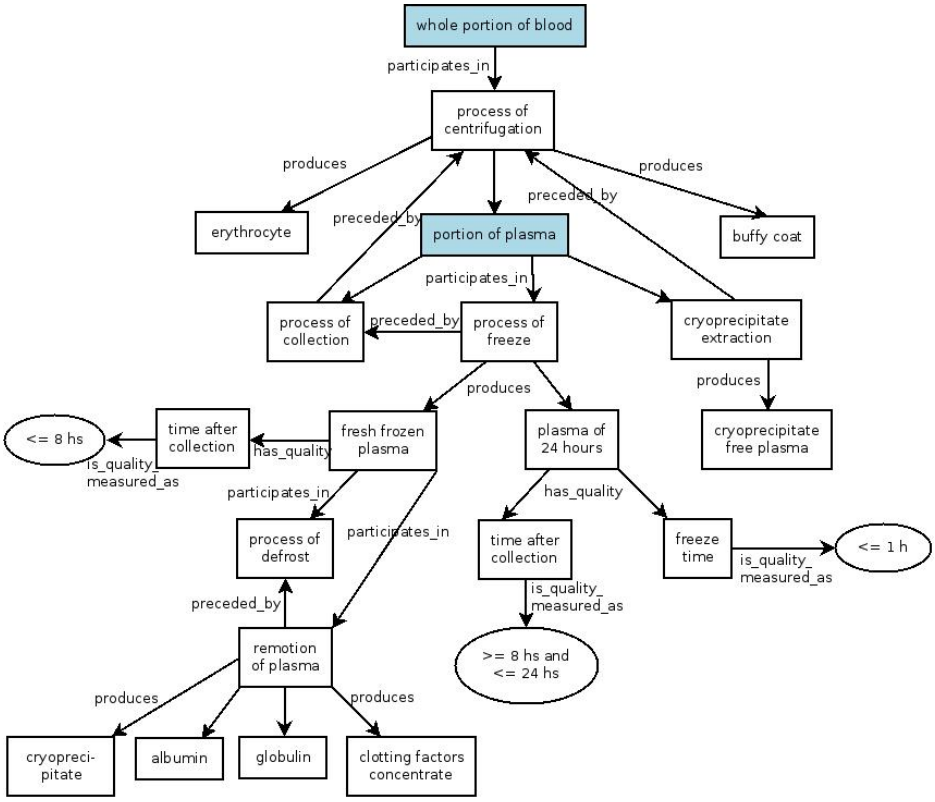


Figure 5: Processes for obtaining of the plasm components.

In order to obtain the hemocomponents *fresh frozen plasma* and *plasma of 24 hours* (left side of figure 5), the initial procedure required is again the *process of centrifugation* of the *whole portion of blood* for separation of *erythrocyte, buffy coat* and *portion of plasma*. The next step consists in the *process of collection* of the obtained *portion of plasma*. The elapsed time after the collection of plasma, here called *time after collection*, is an important parameter in the process as a whole, because it determines the hemocomponent that is going to be generated: (a) when this time is at most 8 hours, the result is the *fresh frozen plasma*, formally, *fresh frozen plasma* **has_quality** *time after collection* **is_quality_measured_as** <= 8 hs; and (b) when the time after collection is between 8 hours and 24 hours the result is the *plasma of 24 hours*, formally, *plasma of 24 hours* **has_quality** *time after collection* **is_quality_measured_as** >= 8 hs and <= 24 hs. In order that both hemocomponents are generated is necessary also that, after the process of collection, the *plasma* be referred to a *process of freeze*. In the case of the *plasma of 24 hours*, the *freeze time* must not exceed 1 hour, formally, *plasma of 24 hours* **has_quality** *freeze time* **is_quality_measured_as** <= 1 h.

The diagram of figure 5 also represents the achievement of the hemocomponents *cryoprecipitate free plasma* and the own *cryoprecipitate*. In spite of the names of these hemocomponents suggest similarities in their respective processes of achievement – extraction of the cryoprecipitate of the plasma and achievement of both -, in practice, this process is performed in a different manner. In order to obtain the *cryoprecipitate free plasma* (right side of figure 5), the initial stage corresponds again to *process of centrifugation* of the *whole portion of blood*, which obtains *portion of plasma, buffy*

coat and *erythrocyte*. The next stage consists in the process of *cryoprecipitate extraction* of the *plasma*. After this extraction, we obtained the hemocomponent *cryoprecipitate free plasma*. On the other hand, the hemocomponent *cryoprecipitate* is obtained from *fresh frozen plasma* (left side of figure 5) with temperature between 1º C and 6º C. This plasma is subjected to *process of defrost* and, then, the supernatant plasma is removed (*remotion of plasma*), leaving in the collection pocket only the precipitate protein and 10-15 ml of this plasma. These products form the hemocomponente *cryoprecipitate*. In this process of *remotion of plasma*, the removed *plasma* **produces** also three hemoderivatives of blood: *albumin*, *globulin* and *clotting factors concentrate*, which are important for blood transfusion.

5. Final considerations and future works

This paper presented a case study in the domain of the human blood describing the construction of an ontology about the human blood hemocomponents and hemoderivatives. We hope that this ontology works as a repository for scientific knowledge about the domain, as well as a instrument to support modeling and evaluation of information systems. In order to achieve this goal, the first stage of the research emcompassed the organization of sets of terms (classes and relations, formally defined) and the creation of representation diagrams (taxonomies, partonomies and other diagrams) to map the knowledge of the studied field.

The next stage of this research will consist in the content evaluation of the ontology by expert professionals. Then, it will be possible the incorporation of new terms and formalisms to ontology. In order to enable the evaluation of the HEMONTO content by experts, we plan to create a web interface to the terms of the ontology with possibilities of searching. This interface will be construct to use the implementation of the ontology in Web Ontology Language (OWL), with use of the ontology editor Protégé. In addition to enable the evaluation of HEMONTO, the search interface will work as a support tool for the biomedical professionals in the learning of specific procedures for blood transfusion.

References

Almeida, M. B.; Proietti, A. B.; Smith, B. and Ai, J. (2011). "The Blood Ontology: an ontology in the domain of hematology". In: *ICBO 2011*; Buffalo, USA.

Almeida, M. B.; Teixeira, L. M. D.; Coelho, K. C.; Souza, R. R. (2010). "Relações semânticas em ontologias: estudo de caso do Blood Project*"*. *Liinc em Revista*, v.6, n.2, setembro, Rio de Janeiro, p. 384- 410.

FMA - Foundational Model of Anatomy [site] (2013). University of Washington School of Medicine, Spokane, Washington, USA. URL: <http://sig.biostr.washington.edu/projects/fm/AboutFM.html>.

Fonseca, F. (2007). "The Double Role of Ontologies in Information Science Research". *Journal of the American Society for Information Science and Technology*, 58, p. 786.

Fonseca, F.; Martin, J. (2007). "Learning the Differences Between Ontologies and Conceptual Schemas Through Ontology-Driven Information Systems". *JAIS - Journal of the Association for Information Systems - Special Issue on Ontologies in the Context of IS,* 8 (2), pp. 129–142.

Grenon, P.; Smith, B. (2004). “SNAP and SPAN: Towards Dynamic Spatial”. *Spatial Cognition & Computation*, v.4, n.1, p. 69-104. URL: <http:// ontology.buffalo.edu/smith/articles/SNAP_SPAN.pdf>.

Grüber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Stanford Knowledge Systems Laboratory, Palo Alto.

Guarino, N.; Giaretta, P. (1995). “Ontologies and Knowledge Bases, towards a terminological clarification”. In: MARS, N. (Ed.). Towards a Very Large Knowledge Bases; Knowledge Building and Knowledge Sharing. [S.l.]: IOS Press, p. 25-32.

Guarino, N. (1998). “Formal Ontology and Information Systems”. In: *FOIS’98*; november 20, 2007; Trento, Italy. Edited by Guarino, N. IOS Press; pp. 3-15.

ICCBBA [site] (2010). “ISBT 128 - Standard Terminology for Blood, Cellular Therapy, and Tissue Product Descriptions”, v 3.33, San Bernardino, CA: ICCBBA. URL: <http://www.iccbba.org/uploads/14/28/14283ab5b4f46c530bd1e57afdefc4f5/Standard-Terminology-for-Blood-Cellular-Therapy-and-Tissue-Product-Descriptions-v4.23.

Masolo, C.; Borgo, S.; Gangemi, A.; Guarino, N.; Oltramari, A. (2003). “Ontology Library:WonderWeb Deliverable D18”.URL:http://www.loa-cnr.it/Papers/D18.pdf>

Rubin, D. L., Shah, N. and Noy, N. F. (2008). “Biomedical ontologies: A functional perspective”. *Briefings in Bioinformatics* 9(1), p.75–90.

Rosse, C.; Mejino, J. L. V.(2003).“A reference ontology for biomedical informatics: the Foundational Model of Anatomy”. *Journal of Biomedical Informatics*, 36:478-500.

Smith, B.; Welty, C. (2001). “Ontology: Towards a new synthesis”. In: Smith B.; Welty, C. (Eds.). *Proceedings of the International Conference on Formal Ontology in Information Systems* (pp. 3–9). New York: ACM Press.

Smith, B. (2003). “Ontology”. In: L. Floridi (Ed.), *The Blackwell Guide to the Philosophy of Computing and Information* (pp. 155-166). Malden, MA: Blackwell.

Smith, B.; Kumar, A.; Bittner, T. (2004). *“Basic Formal Ontology for Bioinformatics”* URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.3787>.

Smith, B.; Ceusters, W.; Klagges, B.; Köhler, J.; Kumar, A.; Lomax, J.; Mungall, C.; Neuhaus, F.; Rector, A. L.; Rosse, C. (2005). “Relations in biomedical ontologies”. *Genome Biology* 6, R46.

Smith, B.; Ceusters, W. (2010). “Ontological realism: A methodology for coordinated evolution of scientific ontologies”. *Applied Ontology* 5, p. 139–188. IO Press.

Soergel, D. (1997). “Functions of a Thesaurus / Classification / Ontological Knowledge Base”. College of Library and Information Services, Univesity of Maryland.

Sowa, J. F. (2000). “Ontology”. URL: <http://www.jfsowa.com/ontology/>.

Spear, A. D. (2006). “Ontology for the twenty first century: an introduction with recommendations”. Saarbrücken, Germany.

Vickery, B. (1997). “Ontologies”*Journal of Information Science*, v. 23, n. 4, p. 277-286.

Wand, Y.; Weber, R. (2004). “Reflection: Ontology in Information Systems”. *Journal of Database Management* 15 (2), iii-vi.

A Navigational and Structural Approach for Extracting Contents from Web Portals

Débora A. Corrêa[1], Ana Maria de C. Moura[2], Maria Claudia Cavalcanti[1]

[1]Department of Computer Engineering
Military Institute of Engineering (IME) Praça General Tibúrcio 80
Praia Vermelha, Urca, Rio de Janeiro, RJ, Brazil

[2]Extreme Data Lab (DEXL Lab)
National Laboratory of Scientific Computing (LNCC)
Petrópolis, RJ, Brazil

deboradac@gmail.com,yoko@ime.eb.br, anamoura@lncc.br

Abstract. *In a semantic Web portal, contents are described and organized based on domain ontologies, and are usually extracted from traditional portals. However, with the increasing amount of information generated each day on the Web, updating semantic portals still represents a major challenge, since this task lacks mechanisms to extract and integrate information dynamically. This paper proposes a strategy to help promoting the interoperability between portals. It consists on the extraction of contents from different Web sites on a specific domain, aiming at the instantiation of a domain ontology, and then use it to update and/or populate a semantic portal. This is carried out through the analysis of the navigational and structural characteristics of traditional portals endowed with some semantic potentiality. In order to evaluate this strategy, a tool named NECOW was implemented. NECOW performance was compared to the Google advanced search mode, and showed promising results.*

1. Introduction

Due to the explosive growth, popularity and heterogeneity of the Web, current traditional portals have difficulties to deal with the maintenance of their pages. They are still very limited for exchanging, reusing and integrating contents of other portals, as well as they rarely present efficient information extraction strategies and metadata maintenance. More recently, many efforts have been devoted in the area of information extraction (IE), whose main goal is to produce structured data from Web pages, so that they become ready for post-processing.

Semantic Portals (SP) arose as an evolution of traditional portals [Brickley et al. 2002][Lausen et al. 2005] [Mäkelä et al. 2004], and emerged as an attempt to provide an informational infrastructure with semantic meaning. They are characterized by the use of ontologies, with the aim of providing more semantic expressiveness to their informational contents. This is achieved by the improvement of some tasks performed over their contents such as search, organization and classification, sharing, publishing and inference. Hence, besides using the same technologies usually used in the construction of traditional portals,

they additionally use ontological languages (RDF[1] and OWL[2]) to better organize structure and provide information semantic meaning in the portal pages [Reynolds et al. 2004].

Despite the advantages of SP, additional techniques are required to ensure these portals can be automatically populated and updated, since many of them still depend on manual update mechanisms. Automatically updating a SP with contents from other traditional portals (sites) depends strongly on Web information extraction (IE) techniques. Due to the heterogeneity and lack of structure of Web traditional portals, access to this huge collection of information is still a challenge, and has been limited to browsing and searching.

Consider, for example, a SP on the education domain that provides information about academic institutions and their courses. When a student wants to collect information about courses from different institutions in Rio de Janeiro, such as UFRJ[3] and IME[4], usually she/he has to navigate through their respective portals. In order to have access to the UFRJ courses, it is necessary to navigate through a list of Web pages, structured in a completely different way from that of IME portal. In fact, this scenario illustrates how difficult it is to extract information from such portals, and consequently, how hard it is to exchange information among them and maintain an up to date semantic portal.

In the literature, some works have been developed in this direction. Makella et al. (2004) use the idea of multi-facets to improve search mechanism in SPs, supported by ontological reasoning capabilities. In [Lachtim et al. 2009], a light ontology on the educational domain is used as the basis for integrating information, developing and populating semantic portals. Although these works aim at enriching portals, and at providing contents with more semantic meaning, they do not contemplate automatic information capturing from other traditional portals (or sites) available on the open Web. In the latter work, an architecture was proposed to retrieve information from semantic Web sites based on domain ontologies, which is then used to integrate contents collected from different SPs. In the present work we extend this idea, since the focus here is on extracting information from traditional portals on a specific domain. This information is transformed into structured data and used to instantiate a domain ontology, which serves as the main basis to automatically instantiate a SP on a specific domain, contributing to its maintenance [Corrêa 2012].

This paper proposes a strategy to deal with the interoperability between portals, also considering the possibility to automatically instantiate a SP. This strategy is based on the instances found along the navigational and structural analysis of Web portals. In order to achieve these goals, we assume that the portals we are going to deal with have some semantic potentiality. This term is used here to refer to traditional portals, whose contents are organized according to a hierarchical structure, helping users to navigate through the subject categories of their interest. These portals, claimed to be potentially semantic, use a somewhat controlled vocabulary, and terms typically appear as links and menu items throughout the portal. Examples of such portals are DMOZ[5], Wikipédia[6], and IME[7]. The

[1] http:// www.w3.org/RDF/.
[2] http://www.w3.org/TR/owl-features/
[3] Universidade Federal do Rio de Janeiro - www.ufrj.br.
[4] Instituto Militar de Engenharia - www.ime.eb.br.
[5] http://www.dmoz.org/
[6] http://pt.wikipedia.org/
[7] http://www.ime.eb.br/

main contributions of this paper are: (a) the specification of a navigational strategy to facilitate the identification of new instances to feed a SP; and (b) the evaluation of the proposed strategy. To the best of our knowledge, it is the first work in the ontology-based IE field that follows a navigational strategy.

The remainder of this paper is structured as follows. Section 2 gives a brief description of some essential concepts that are used throughout the paper. Section 3 presents some related work. Section 4 describes our navigational strategy for automatically extracting contents from portals and populating an ontology, with a brief description of its main functionalities. Section 5 presents NECOW, an extraction tool that has been developed according to the strategy proposed, with an example to demonstrate its usage. Section 6 is dedicated to the tool evaluation, and finally section 7 concludes the paper with suggestion for future work.

2 Extracting Information from Web Portals with Semantic Potentiality

Some traditional portals do organize their contents according to a hierarchical structure, helping users to navigate through the subject categories of their interest. However, in this work, we develop a navigational strategy to extract information based on structures found on portals that present some *semantic potentiality*. We define such a portal as the one that contains one of the following characteristics: (i) has links, lists or tables and benefits from any kind of organization and hierarchy in its structure; and/or (ii) some of its pages are presented as a taxonomy, although not all of them.

While DMOZ and some academic portals such as those of IME and UFRJ are classified in this category, others such as DBLife[8], DBPedia[9], FreeBase[10] are considered more comprehensive collaborative portals, since they provide a wide set of services that help dissemination and sharing information.

Semantic portals make use of semantic Web technologies to improve important functionalities in a portal, such as search and organization. Among these technologies, ontologies are considered as the most significant ones, since they enable common understanding and sharing of a domain between humans, agents and applications. Ontologies are also crucial to organize SPs, grouping sites and documents in pre-defined sets, according to their contents.

Due to the great heterogeneity of structures embedded in Web pages, extracting relevant data from them is still a challenge. IE is a classic text mining technique, whose goal is to find some specific information in texts, by identifying information contained in non-structured information source. This information should be in agreement to a predefined semantics, so that it could be later stored and/or manipulated by several other sources.

In the literature, three important IE techniques are identified [Silva A.S. 2012]: i) wrappers; ii) those based on Natural Language Processing (NLP); and iii) those based on the Deep Web (DW). The first one aims at extracting information from structured or semi-structured data (such as HTML). They are based on their format, delimiters, typography and frequency of words. NLP aims at extracting information directly from unstructured texts, and depends on the natural language pre-processing such as in Ondux [Cortez et al. 2010] and JUDIE [Cortez et al. 2011]. Finally, those based on the DW aim at extracting

[8] http://dblife.cs.wisc.edu
[9] http://dbpedia.org
[10] http://www.freebase.com

information from forms and/or hidden tables that are not visible to the user, as in DeepPeep [Barbosa and Freire 2005] and DeepBot [Álvarez et al. 2007].

Wilmalasuriya and Dou (2010) wrote an interesting overview about ontology-based IE technologies, also exploring some related tools. However, to the best of our knowledge, none of the discussed works used a Web navigational strategy, nor focused on the maintenance of semantic portals.

3. Related Work

A challenging research topic for the Web researcher's community is the interoperability between portals and their automatic instantiation. The literature points out to some works that use semantic Web technologies to exploit this topic, although in different contexts.

Lachtim et al. (2009a, 2009b) created an educational semantic portal, which is populated and integrated with contents extracted from Web semantic pages within the same domain. In [Suominen et al.2009] metadata and documents are obtained from contents published in Content Management Systems or from those manually annotated by the metadata editor SAHA [Kurki and Hyvönen 2010]. Later these metadata are submitted to an ontology to be validated and published in a semantic portal. The portal presented in [Hyvönen et al. 2009] creates its contents by making use of a set of metadata schemas and some specific tools. This population process enables producing and extracting contents from museums, libraries, files and other organizations, besides getting information from citizens as individuals and from national and international Web sources.

When compared with these works, our great differential consists on the semantic portal update with contents hosted in sites and/or Web portals with some semantic potentiality, and considering only their presentation and navigational structure, such as links, lists and tables. Hence, the update task in these portals allows these pages to be transformed from simple user pages into ones that are able to integrate and instantiate contents based on domain ontologies.

4. An Approach for Navigating and Extracting Information

This work extends the architecture proposed by Lachtim et al. (2009a). The latter aimed at creating a semantic portal, integrating and instantiating a domain ontology that supported a SP with contents extracted from Web semantic pages within the same domain. However, that architecture did not consider contents extracted from traditional portals or sites in the open Web. This work proposes a strategy to fill in this gap, as described along this section.

Figure 1 presents an overview of the proposed strategy. Mainly, the idea is to navigate through a list of sites with some semantic potentiality, on a specific domain. The navigation is guided by a subset (cropping) of a domain ontology (OB), which is represented in OWL. For each site in the list, *useful*[11] information is extracted to enrich the ontology, i.e., new potential instances of the OB classes, as well as new potential relationships between them (instances of OB object properties), are identified. A user validation of such new information is needed in order to remove eventual false positives. All this information is then transformed into RDF triples, which compose a new version of the OB ontology, here called OB´. The OB´ ontology may be used as input for the alignment with the already existing information in the current semantic portal. The main component of

[11] In the context of this paper, *useful* means all kind of information that is pertinent with the current domain.

this architecture concerns the IE, which gives the required support for populating portals. This component, as illustrated in Figure 1, is composed of modules, whose characteristics and functionalities are described next.

i.OB cropping: this step is responsible for loading a list of classes, instances and properties of the OB domain ontology, which will be the basis for the search of the *useful* instances of each page visited in a portal with semantic potentiality. The relationships between classes of the OB are also considered, since they guide the navigation along the portal pages. The navigation always starts from the most general class, defined by the user, and proceeds to the more specific ones. Additionally, the real name of each class, its label and its equivalent classes are very important for the navigation between the pages of the portal (see step iii). The instances of each class, as well as their equivalent instances (defined by the property *same as*) are also considered;

ii.Pre-categorization and identification of the initial page: a pre-categorization based on the title will be performed to limit the navigation defined by the step (i). If the initial page contains in the title a name similar to an instance of any OB class, the navigation will start from the next class of the OB. If this situation does not occur, the navigation will start from the first OB class;

iii. Navigation: this module is responsible for the navigation through the pages of each site previously defined by the user (stored in a configuration file). Its main goal is to search for links, within table or menu lists, through which OB' classes can be identified and corresponding new instances can be retrieved. It is composed of four sub-modules:

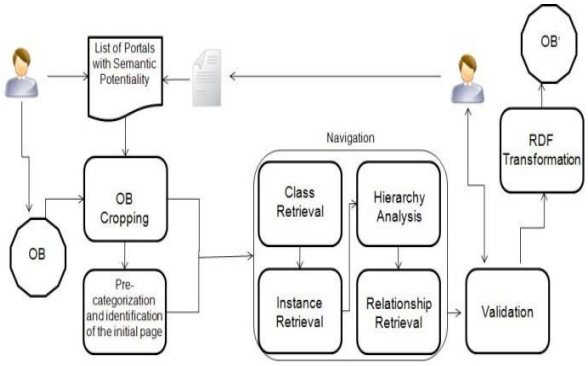


Figure 1. Navigation and Information Extraction Module.

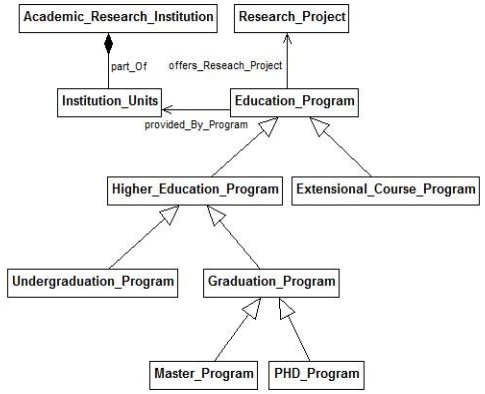


Figure 2. Fragment of the OBEDU ontology (OB)

A.Class retrieval: once the navigation starts, the system will search for links and labels that are similar to the desired OB' classes defined in step (i). These links will be considered as priority for navigation. Whenever a similar link is identified, the system verifies if it has already been visited. In the affirmative case, it will go on through the next link; otherwise, the link will be visited and its instances will be retrieved as described in the next step B;

B.Instance retrieval: for each OB' class similar link identified in step A, the corresponding target page is traversed in order to identify potential instances to that class. These instances should appear between *tags*, denoting links, lists and/or table items. Additionally, their label should have some similarity with the existing instances of the

corresponding OB' class. During the navigation, all the information extracted is saved for later validation by the user (step iv);

C.Hierarchy analysis: in order to avoid duplicated instantiation in the OB´, hierarchies should be verified. This duplication typically occurs, for example, with a class and its subclasses. As an instance can instantiate a class and also its superclasses, the most specific one is chosen;

D.Relationship retrieval: associations between instances should be in accordance with the existing OB relationships. Hence, for example, in the ontology shown in Figure 2, the instances of "*Education_Program*" are associated with those of "*Academic_Research_Institution*" through the property "*provided_By_Program*". Among the new set of instances, such new associations are also identified, and later transformed into RDF triples (step v);

iv. Validation: this module is responsible for allowing the user to validate all the information extracted by the system during navigation. Even that one that may be considered as invalid is also saved, in order to be used later in a pre-validation process. This information can be confronted with the one that is retrieved later, during a posterior navigation;

v. RDF transformation: this module converts valid information into RDF triples, which will be included in the new ontology, the OB´. Actually, this corresponds to an empty crop of the OB, which is updated with the new instances extracted during navigation. OB´ triples can be submitted to an ontology alignment process with the OB ontology, and its instances will then be used to populate a semantic portal having the OB as its domain ontology. This alignment step is not in the scope of this paper.

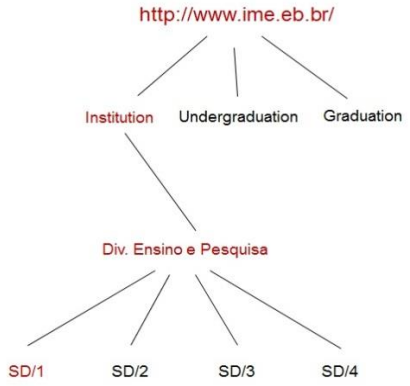
5. NECOW: a Prototype Tool

This section describes the prototype tool, named NECOW (Navigation and Extraction of COntents on the Web), developed with the objective to evaluate and test the strategy proposed in this work. It is a Web friendly tool developed in Java 1.6, and supported by some libraries (Jena[12], Jericho parser HTML[13], etc.). Navigation in NECOW starts from a portal Web link defined by the user, with the support of the base ontology (OB), which is loaded in memory and will help during all the navigation process for the search of classes and instances. It is worth observing that the strategy presented in section 4 is a generic proposal, and may be applied to other domains, for which there is a domain ontology. However, in order to show how this strategy is performed using NECOW, we will use an example in the educational domain, which is supported by the OBEDU ontology [Lachtim et al.2009a], which provides English and Portuguese vocabulary. A fragment of this ontology is presented in Figure 2. We also start our use case example with the IME institution, described through its portal, as shown in Figure 6.

When navigation starts through this portal, the html page source code of each page visited along the process is analyzed to verify if the label corresponding to the tags *title, link, item list* and *HTML tables* (*<title>,<a>, * and *<td>*, respectively) has any similarity with a class and/or instance of the OB ontology. When this occurs, the corresponding tag labels are stored in a list. Hence, this navigation follows the same principle of a crawler, where only links associated with the OB are used in the process. Additionally, each candidate instance selected is stored in a list associated with its class

[12] http://jena.sourceforge.net/ontology/ - used to manipulate ontologies.
[13] http://jericho.htmlparser.net/docs/index.html -used to extract information from Web pages during navigation.

(Figure 3). However, for the tag<*a*> the process is different: the *href* (attribute that contains an *url*) and the tag *label* are extracted. For the labels of the tag<*a*> the same analysis for the labels is done, while for the *href* content, the links containing a label or a word (in its own kink) that has any OB class are extracted and stored in a list of links. Later these links are structured dynamically as an *n-ary* tree to identify the navigation path and the relation between these links.



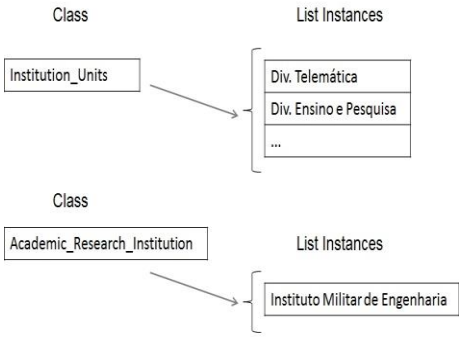


Figure 3. Instances of each class **Figure 4. A Tree representing the portal links**

This tree also indicates which instances have been extracted for that link and their original classes. Figure 4 presents the navigational structure of the IME portal, whose nodes correspond to the links found and that will be visited during the NECOW execution.

In Figure 6, the first link is followed by the *href content http://www.ime.eb.br*. When accessing the page referenced by this link, we find "Instituto" (Institution, in English) as the content of the *href http://www.ime.eb.br/index.php?option=com_content&view=article&id=219&Itemid=3*. Performing again the previous step, the link "Instituto" directs us to the link "Div. Ensino e Pesquisa, SD1" (SD1 is the name of a Teaching and research division at IME), etc. (Figure 5). Navigation is performed in a *breadth-first* search, and before storing a link the system verifies the category of the link: if it is identified as a relative link [14] it is concatenated according to its domain, otherwise the absolute[15] ones are stored integrally. The similarity degree between words is calculated according to the edition cost, using the Levenshtein algorithm [Navarro 2001]. The edition cost consists on obtaining the number of operations (insertion, delete or modification) required to transform a word into another, one character at a time. This cost comprehends an interval between 0 and the size of the biggest word. Zero indicates the words are the same, and the larger the value, the greater the number of operations performed, and consequently, more different the words are.

During navigation, the associations found between the links and their respective lists and tables (HTML) are compared with those of the OB, as illustrated in Figure 7. Those that are in accordance are stored, and at the end of the navigation they are presented to the users as RDF triples to be validated. When the associations are not identified during navigation, the relationships based on the ontology associations are suggested to the user.

[14] Its address is written in a summary way, containing only their directory names.
[15] The address is written integrally.

6. Evaluation and Results

This section is dedicated to the evaluation of the NECOW tool. Although the proposed approach is generic and can be applied to any specific domain, our test scenario has been developed on the educational domain, according to the ontology partially described (OBEDU) (Figure 2). The tests applied to NECOW aim at evaluating the tool efficiency with respect to the data extracted from the Web, in terms of precision and recall measurements. The extraction results were then analyzed and some were considered as valid instances to update the POSEDU portal [Lachtim et al. 2009b].

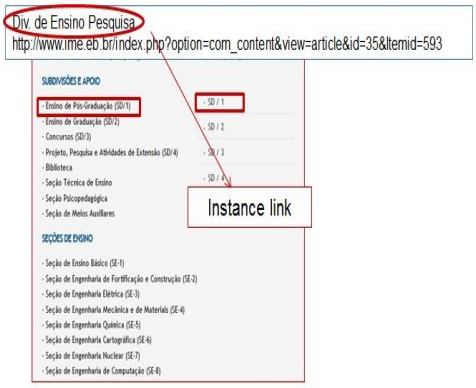


Figure 5. Navegational structure of the IME portal

Figure 6. IME portal and links to the tree of links

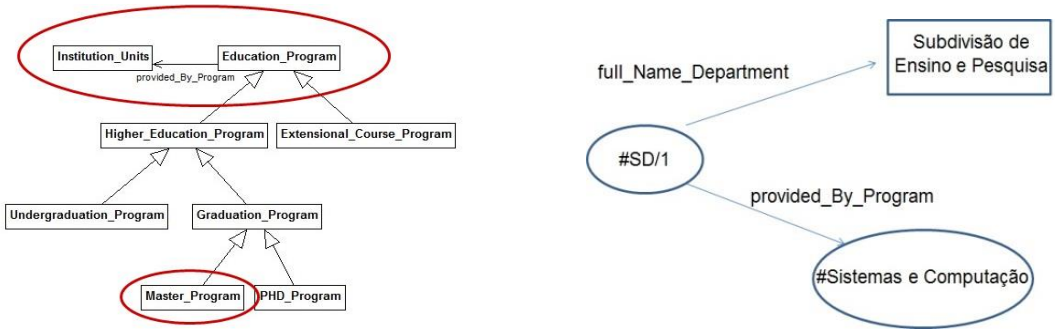


Figure 7. Instances found at IME portal related to the OB ontology

6.1. Test Scenario

At the time of developing this work we did not find in the literature a navigation tool with similar navigation strategy as NECOW, in order to use it as a comparison platform to evaluate our tool efficiency concerning its usefulness. Therefore, we evaluated NECOW navigation results by comparing them with those obtained from Google advanced search mode, over the same set of sites. In order to calculate the results precision and recall for each tool, we considered navigation and information extraction performed manually by a set of users, as the *gold standard reference*. Manual navigation was performed by a group of 20 users. Each user was told to select manually all the information considered relevant from a given list of portals. Both NECOW and Google also browsed the same list of portals. These portals were previously selected, taking into account that they all belong to a similar domain. Additionally, a related domain ontology (OB) was also chosen. It is worth observing that the automatic selection of portals (with semantic potentiality) on a specific

domain is not in the scope of this paper. We assume these sites have been submitted to a previous analysis, to ensure they had some semantic potentiality, and that they could contribute with useful information for the OB ontology. Navigation was accomplished the same way in all procedures (manual, NECOW and Google), taking into account the same order in which classes and relationships were organized in the ontology.

The manual procedure was performed as follows. Initially we distributed a guide to the users containing a list of items to be considered, which includes: a list of links of educational portals to visit; a starting point page for each link in the visiting list; a list of the ontology terms they should search in each link; a list of similar terms to be used in an extended search expression. Additionally, while following such instructions, each user filled in a form with information concerning the ongoing manual navigation, such as: the page in which he/she was navigating; the terms (generic and /or specific) used to arrive at that page, as well as the path used; and also if these terms were linked to other pages. At the end, a set of new instances to the base ontology classes were documented by such users, and these were taken as the *gold standard reference*. With respect to the navigation procedures with tool support (Google and NECOW), the result set was compared to the *gold standard reference*. Navigation and information extraction executed by Google starts with the submission of a search expression built based on each ontology class (and its equivalent classes). Then, besides analyzing the returned page with a list of URLs, the user all the pages pointed by each URL are also analyzed. The search expression that was submitted to Google was manually built as follows: a class name followed by its equivalent classes separated by the OR operator, and followed by the (initial) link site. An example of one of the used expressions is: *Institution Unities OR Institutes OR Faculties - http:ime.eb.br.*

NECOW navigation and information extraction starts with a given web page (e.g. http://ime.eb.br), as explained in detail in section 5. Different from Google search mechanism, NECOW crawls through the site with the help of a chosen ontology structure (also a user input choice), and the user does not need to navigate through links. The candidate instances are suggested by NECOW at the end of its execution. Our tests have been performed over twenty specific sites on the educational domain, corresponding to some Brazilian universities. This number of sites was defined based on [Lachtim et al. 2009a] and [Navarro 2001]. Both works describe similar experiments concerning manual navigation, whose evaluation process is hard and tiresome, since it can generate a large number of instances. Based on preliminary tests, we considered as candidate instances (i.e., the ones that might be useful to be included in the portal) those that presented a similarity degree between 0 and 0.5, a value calculated according to the edition cost algorithm.

The experimental tests aimed at comparing NECOW results with the results obtained from Google, taking into account the *gold standard reference*, i.e., the results obtained with the manual navigation procedure. The set of results were the base to calculate the *precision* (P), *recall* (R) and *F-Measure* (F) coefficients, defined respectively by: $P = \frac{|Ra|}{|A|}, R = \frac{|Ra|}{|Ri|}, F = 2 * \frac{R*P}{R+P}$, where: Ra corresponds to the number of the relevant information instances retrieved by either NECOW or Google; A is the number of the information instances retrieved respectively by each tool; and Ri is the number of relevant information instances obtained with the *gold standard reference*.

6.2. Evaluation and Discussion

The portals were grouped according to the recall values, due to what was observed during preliminary tests: when we decreased the superior similarity threshold value, aiming at increasing precision, this one and the recall itself decreased considerably. Besides, many valid results were returned as invalid (false negatives). A more detailed analysis of the invalid results (true negatives at first) showed that some of them could be possibly evaluated as relevant if confirmed by the user. Since these *doubtful* results do not influence recall, but only precision, they have not been classified as invalid, in the invalid list.

The categories defined for recall have been defined as: (1) High recall (HR): results defined in the interval [0.70, 1]; Average recall (AR): results defined in the interval (0.40, 0.69]; Low recall (LR): results defined in the interval (0, 0.40]. Figure 8 shows the unified results obtained from Google and NECOW. Taking into account the *gold standard reference*, NECOW obtained the best recall results for most portals (UFJF, UFMG, PUC-Rio, UNESP, UFLA, UFF and UFG), whereas it presented the same results as Google for the other portals. A brief analysis of these portals, also considered as well *behaved portals* (HR), lead us to conclude that most of them present a good navigation and presentation structure, i.e., those that follow the basic HTML best practices. Furthermore, the terms used by them to describe the domain are quite close to those present in the OB. This explains the great difference observed with UNESP university portal, which reached 0,91 from NECOW, and 0,18 from Google. Additionally, we also observed the presence of many valid links and well defined labels in the portals, such as in IME, UFMG, UFRGS, UFC, UFJF and PUC-Rio universities. Considering the categories defined above and the lower recall obtained by both tools, we remarked that some portals have been classified differently by NECOW and Google. Similar situations occurred to UFLA (0.58) and UFG (0.46) that were classified as AR by NECOW, but as LR by Google (0.16 and 0.21, respectively); and to UFF classified as LR by NECOW (0.30) and as AR (0.47) by Google.

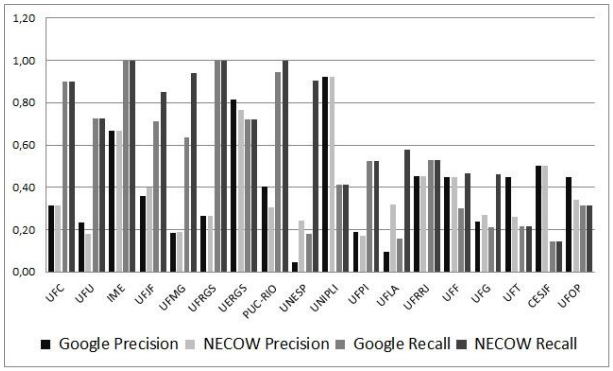


Figure 8. NECOW and Google results compared with the *gold standard reference*

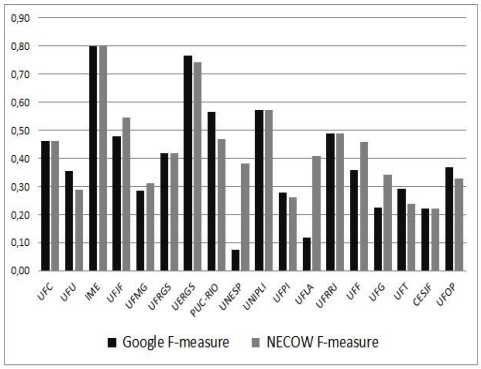


Figure 9. NECOW and Google *F_measure* results

A detailed analysis of these values may justify such low results: presence of some portal internal links that redirects the server link, taking NECOW to invalid links; links that are not similar nor equivalent to the terms of the OB; terms that do not follow their standard usual connotation, or even links without any associated page; some information in the domain context were available through text indentation (TABs), difficult for the NECOW parser to find the desired information; use of <*link*> tag, instead of <*a*> tag, usually used for style sheets that, in conjunction with the use of frames, are not considered by NECOW yet. Two of our portal list fell in these cases. Precision results were lower than the recall ones. NECOW presented a discrete advantage in comparison to Google (UFJF, UNESP, UFLA

and UFG), while for the others they presented similar results, except for UNIPLI, where we can observe a precision improvement (0.91). Actually, this portal presented a very few number of informational contents compatible with the OB concepts and hence, the information captured by NECOW was worse. Figure 9 presents the *F_measure* graphic. Its maximum values corresponded to the portals that had also the highest recall values (IME, UFRGS). From the analysis carried out along this work we concluded that NECOW obtained best recall results than Google. Additionally, it was possible to list some characteristics of portals with semantic potentiality, according to their classification into high, average and low categories, summarized in Table 1.

Table 1. Classification of Portals according to their Semantic Potentiality (SP).

	High SP (0.7 <=R<=1.0)	Average SP (0,4<= R <=0,69)	Low SP (0.0 <=R<=0.39)
Good presentation and navigation	Yes	Yes	No
Used terms similar to the ontology	Many	Acceptable	Few
Terms probably require padronization	No	Acceptable	Few
Use basic HTML tags	Yes	Yes	No
Tag "inside" Tag	Few	Few	Many
Use other kind of tags	No	Yes	Yes

7. Conclusion

This paper extends a previous work [Lachtim et al. 2009a], by proposing a strategy to collect contents from sites and/or traditional portals with some semantic potentiality within a specific domain, in order to instantiate an existing domain ontology that supports a semantic portal in this same domain. Actually, this strategy aims to facilitate the population and update procedures of semantic portals. In order to test and evaluate this proposal, a tool (NECOW) was implemented, and some tests were performed comparing it with Google advanced search tool, having as reference set the manual navigation performed by a group of users. It was possible to observe that manual navigation is usually more precise, and that the lack of structure in many portals design turns navigation and automatic extraction very hard. However, the good recall results obtained with NECOW were promising. It may be considered as an interesting and powerful tool to complement other IE techniques based on natural language processing, in the attempt of (semi) automatically populating semantic portals. As future work we intend to use machine learning techniques to improve information extraction process, as well as test other algorithms to calculate similarity between strings during this process.

Acknowledgements. This work has been partially supported by CAPES graduation scholarship and by CNPq through its Institutional Capacity Program (Proc. 382.489/09-8) and Productivity Research fellowship program (Proc. 309307/09-0).

References

Álvarez M., Raposo J., Pan A., Cacheda F., Bellas F., Carneiro, V. (2007). DeepBot: A Focused Crawler for Accessing Hidden Web Content. *University of La Coruña*.

Brickley D., Buswell S., Matthews B. M., Miller L., Reynolds D., Wilson M.D. 2002. Semantic Web Advanced Development for Europe (SWAD-Europe). In Proc. of the *1st Int. Semantic Web Conf. on The Semantic Web*, Sardinia, pages 409-413, 2002.

Barbosa L., Freire J. (2005). Searching for Databases. 18th *International Workshop on the Web and Databases* (WebDB 2005), Baltimore, Maryland.

Corrêa D.A. (2012). An Approach for Extracting Contents Based on Structural and Navigational Characteristics of Web Portals (in Portuguese). *Master thesis*, IME, Rio de Janeiro, Brazil, April.

Cortez, Silva A., Moura E. (2010). Ondux: On Demand Unsupervised Learning for Information Extraction. Proceedings of the ACM SIGMOD International Conference on Management of Data.

Cortez, Silva A., Moura E., Laender A. (2011). Joined Unsupervised Structure Discovery and Information Extraction. *Proc. of the ACM SIGMOD Int. Conf. on Mmt. of Data*.

Hyvönen E., Mäkelä E., Kauppinen T., Alm O.,et al. (2009). CultureSampo - Finnish Cultural Heritage Collections on the Semantic Web 2.0. *Proc. of the 1st Int. Symposium on Digital Humanities for Japanese Arts and Cultures (DH-JAC-2009)*, Ritsumeikan Univ., Kyoto, Japan, March.

Kurki J., Hyvönen E. (2010). Collaborative Metadata Editor Integrated with Ontology Services and Faceted Portals. *Workshop on Ontology Repositories and Editors for the Semantic Web, the Extended Semantic Web Conference ESWC*, Heraklion, Greece, CEUR Workshop Proceedings.

Lachtim F. A., Moura A. M. C., Cavalcanti M. C. (2009a). Ontology Matching for Dynamic Publication into Semantic Portals. *Journal of Brazilian Computer Society* (JBCS), ISSN: 0104-6500, vol 15. pp 27- 43, Mar.

Lachtim F. A., Ferreira G.N., Gama R., Moura A. M. C., Cavalcanti M. C. (2009b). POSEDU: a Semantic Educational Portal. *IEEE Multidisciplinary Engineering Education Magazine*, vol 4, nº 3, pp. 65-75, ISSN:1558-7908.

Lausen H., Ding Y., Stollberg M., Fensel D., Hernandez R., Han S. (2005) Semantic Web Portals: State-of-the-Art Survey. *J. Knowledge Management*, V.9, N.5. May, pp. 40-49.

Mäkelä E., Hyvönen E., Saarela S., Viljanen K. (2004). Ontoviews - a Tool for Creating Semantic Web Portals. *Inte. Semantic Web Conference*, Hiroshima, pp. 797-811.

Navarro G. (2001). A guided tour to approximate string matching. Univ. of Chile. ACM Computing surveys, vol. 33, no. 1.

Reynolds D., Shabajee P., Cayzer S. 2004. Semantic Information Portals. *ACM*, NY, May.

Reynolds D., Wilson M.D. (2002). Semantic Web Advanced Development for Europe (SWAD-Europe). *In Proceedings of the 1st Int. Semantic Web Conf. on The Semantic Web*, Sardinia, pages 409-413, 2002.

Silva A.S. (2012). Methods and Techniques for Information Extraction by Text Segmentation. *Proc. of the 6th Alberto Mendelzon International Workshop on Foundations of Data Management*, Ouro Preto, Brazil, June 27-30.

Suominen O., Hyvönen E., Viljanen K., Hukka E. (2009). *HealthFinland - a National Semantic Publishing Network and Portal for Health Information*, Finland.

Wimalasuriya D.D. and Dou D.(2010). Ontology-based Information Extraction: an Introduction and a Survey of Current Approaches. J.Inf. Sci. 36, 3, June.

An Ontology Reference Model for Normative Acts

Pedro Paulo F. Barcelos[1], Renata S. S. Guizzardi[2], Anilton S. Garcia[1]

[1]Electrical Engineering Department – PPGEE
[2]Department of Computer Science - PPGI

Federal University of Espírito Santo - UFES
Vitória – ES – Brazil

pedropaulofb@gmail.com, rguizzardi@inf.ufes.br, anilton@inf.br

Abstract. *Normative Acts are important legislative and regulatory documents made by different governmental organs. Every year, a huge amount of information is provided in Normative Acts by these organs without control, i.e., there is no effective way to verify redundancies, inconsistencies, cross-impact and ambiguities. In this paper, we propose a domain ontology for Normative Acts based on official documents (the Brazilian Constitution and the Redaction Manual of the Presidency of the Republic) as a reference model that can be used to improve communication, interoperation and automation of Normative Acts. The reference model is built with a highly expressive well-founded language within a methodology that ensures its quality.*

1. Introduction

Among the various duties of the Brazilian powers, one of its main activities is the publication of Normative Acts (NAs) to establish standards and to inform decisions and other information to society. The main types of Brazilian NAs can be found in Article 59 of the 1988 Constitution of the Federative Republic of Brazil. NAs are central to legislative power, where these acts are daily created in the form of laws. They are also important to the executive power, where they are mainly created in form of executive decrees. In addition, they are essential to the national regulatory agencies, where they are created in the form of resolutions.

The activities done by these organs to elaborate, edit, and publish NAs are complex and involve several organizational units and a large number of human resources. Different people, with different cultural and technical knowledge and with different interests are involved with the creation of NAs. Thus, these people must share a common comprehension about the terms and concepts related to NAs, in order to improve the resultant document. Due to the legislative and regulatory importance of NAs, miscomprehension of concepts during the planning and elaboration of NAs can range from a simple structure error (resulting in a difficulty to automatically read the generated document with an computational application) to a huge interpretation problem, generating social and financial losses to society and companies. Moreover, to be published, the NAs text must be clear and unambiguous, as society and other public and private organs have to comprehend and share (textually or computationally) its contents.

In order to produce the desired impact to society, the NAs must be carefully planned in their elaboration stage, involving studies and researches of previous related NAs. Today, these studies and researches are done manually, without intelligence provided by computational applications – to research a NA, an editor must use a non-specialized research tool, just like any layman would. Moreover, once published, its content must be easily accessible and researchable by society and other stakeholders. In fact, in Brazil, a federal law ensures that every public organ or entity must publish, in detail, the formats used to structure their information. It is widely known that, although a reference document exists for the NAs writing, this document is not always used by the legislative houses and regulatory agencies, thus resulting in the above cited problems.

The official reference document which deals with the NAs' writing is the Presidency Writing Manual (in Portuguese, *Manual de Redação da Presidência da República* – available at http://www.planalto.gov.br/ccivil_03/manual/). The Writing Manual is divided into two parts: the first, among other topics, presents the official communications and standardizes the layout of expedients. The second part deals with the elaboration and wording of NAs and presents examples of the Normative Acts and the legislative procedure.

This second part of the document, in particular, describes the textual elements that a NA may have and their inter-relationships, making it a valuable reference and a natural candidate for the development of an Ontology Reference Model for this domain. Even though the description contained in the Presidency Writing Manual is available only in natural language (Portuguese), which does not guarantee absence of ambiguities, the use of an ontologically well-founded language can identify and correct these possible deficiencies. Such type of language differs from other commonly used languages to represent data or knowledge in information systems (like databases schemas) as they are built accordingly to a foundational ontology, i.e., a meta-ontology that describes a set of real-world categories that can be used to talk about reality [Guizzardi 2007]. A framework for ontological evaluation is presented in [Guizzardi 2005], and an example of an application of this kind of evaluation in a network language, also described in natural language, can be seen in [Barcelos et al. 2011].

Our objective in this paper is to present a domain ontology developed to be a reference model for NAs. Reference Models are essential artifacts when dealing with information of a given domain, as they formalize concepts and their relations in a clear and unambiguous way, improving communication and information exchange and interoperation. In fact, the main objective of a reference model is to assist humans in tasks such as meaning negotiation and consensus establishment. This goal can be achieved by using highly expressive languages, within a formal ontology engineering methodology, to create a strongly axiomatized ontology that approximates as well as possible to the domain conceptualization. The focus on these languages is on representation adequacy, instead of computational representation [Guizzardi 2007]. The Normative Acts ontology reference model is formalized with OntoUML, an ontologically well-founded profile of the Unified Modeling Language (UML). The ontology engineering methodology used guarantees the validity and correctness of the modeled information through syntactical and semantic validations.

In brief, the main contribution of this paper is to provide an ontology reference model for the NAs domain, formalized with OntoUML within an ontology engineering methodology. A secondary contribution is the presentation of this ontology`s capabilities to be used as basis for computational implementation.

The work described here is placed in the context of the Information and Knowledge Management Model Project ("Modelo de Gestão da Informação e do Conhecimento", MGIC, in Portuguese), a cooperation project between the Federal Fluminense University and the National Agency of Terrestrial Transportation (ANTT). The MGIC aims to improve efficiency in ANTT's decision-making. To achieve this goal, the information modeling work is performed on four different fronts. The ontology modeling, one of these fronts, aims to create conceptual reference model ontologies for structuring the agency's information.

This paper is structured as follows: in section 2, we compare this work with others that are related to NAs; section 3 presents the ontology engineering methodology used to create and guarantee the quality of the proposed NAs ontology; in section 4, we describe the ontology reference model for NAs in details; section 5 presents possibilities of practical use of the NAs ontology as a conceptual model and as basis for computational implementation; section 6 presents the main conclusions as well as future works.

2. Related Works

As the legislative process involves several public entities in various spheres and as it generates a large amount of information, a huge number of works are published involving the creation of ontologies used for communication and automation in this domain. In fact, some of these works are the results of large projects, like the LexML Project [Lima 2010], and the International Ontojuris Project [Clara et al. 2010].

Legal ontology projects differentiate from each other in scope, objective and in which kind of ontology they are based. As an example, the Brazilian LexML project uses as basis an ontology called FRBROO, an ontology in the domain of cultural heritage; while the Ontojuris Project uses a lightweight ontology (almost a dictionary), similarly to other initiatives, such as the LTS (Legal Taxonomy Syllabus) [Ajani et al. 2009], in Europe.

The Power project, in its turn, uses shared conceptual models to facilitate the legislation process [Engers and Glassée 2001]. [Visser and Bench-Capon 1996] also proposed a Legal Ontology specification, while [Boer et al. 2003] proposed an ontology for comparing and harmonizing legislation. Other works aim to provide foundations for concepts of law, like [Breuker and Hoekstra 2004]. These works and the ontology proposed in this paper, however, do not share the same scope, as the foundations are provided to models representing legislative process concepts, not the internal elements and relations of Normative Acts.

Even though there is a vast number of works about ontologies in the legislative area, it appears, however, that no work involves the creation of a well-founded ontology reference model for the representation of the Normative Acts' internal structure.

3. Ontology Engineering Methodology

In order to create an ontology reference model that correctly reflects the intended domain and that is able to be used by different agents (people, groups of people and other, like machines) to interoperate, an ontology engineering methodology must be used. This section presents our methodology, partially based on the Ontological Approach to Domain Engineering presented in [Falbo et al. 2002]. In our methodology, shown in Figure 1, we use the steps of the Ontological Approach to Domain Engineering with different level of rigor, abstracting non-essential elements to our case.

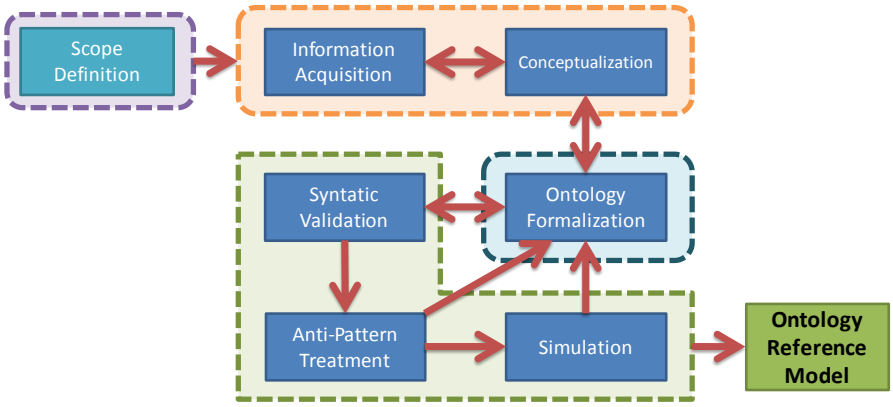


Figure 1 – The used Ontology Engineering Methodology

Scope definition is the first step of the iterative methodology. Our ontology uses a reference document, the Presidency Writing Manual, written in natural language (Portuguese in our case), as the modeling scope.

The second step of the methodology is the *ontology capture*, were the sub activities of Information Acquisition and Conceptualization are realized. In order to acquire information, a domain study is necessary for the modeler to learn about the subject to be modeled. We used the Presidency Writing Manual and the Brazilian Constitution as main source of information. Conceptualizations are immaterial entities that only exist in the mind of the user or a community of users of a language. In order to be documented, they must be captured in terms of some concrete artifact. This implies that a language is necessary for representing them in a concise, complete and unambiguous way [Guizzardi 2007].

The *ontology formalization* step consists in the formalization, through diagrams, of the domain model. In order to correctly represent a domain, an expressive language must be used. This language should be able to represent information despite of implementation technologies or limitations. In this work, we use OntoUML, an ontologically well-founded UML profile [Guizzardi 2005]. As graphical languages are not always capable of correctly representing the domain, some Object Constraint Language (OCL) rules are also necessary for restrictions and derivations rules.

As we intend to create a reference model for Normative Acts (NAs), it is important to ensure that the diagrams allow only instantiation as desired, that is, that the user can only create instances that are possible in the real world. To do this, we focus on the *validation* of information modeled at the diagrams. In this stage, we have two main types of validation: (1) the syntactic one, which guarantees that the OntoUML models

created are syntactically correct, that is, that the entities created are according to the languages meta-model; and (2) the semantic validation, where we want to avoid syntactically correct diagrams that can be instantiated to generate undesired world of affairs.

The OntoUML Lightweight Editor (OLED) (https://code.google.com/p/ontouml-lightweight-editor/) provides the *syntactical validation*. The Semantic Validation does not ensure that there is no impossible state of affairs allowed by the ontology, it does, in fact, ensure that its occurrences are reduced. The semantic validation is done in two steps, within an OLED's module, called MOVE (OntoUML Model Validation Environment): the first step is an *anti-patterns identification* and treatment and the second step is a *simulation* using Alloy.

As stated in [Sales et al. 2012], an anti-pattern is a recurrent decision for a specific scenario that usually results in more negative consequences than positives ones. The MOVE tool provides a model verification to check occurrences of anti-patterns. Simulation can help the modeler to find inconsistencies and unwanted worlds of affairs allowed by the model. The MOVE tool can translate the model to Alloy [Jackson 2002]. Alloy is a model-checking language that can be used to simulate possible worlds based on the formalization provided. This kind of validation guarantees the validity of modeled information inside an specific context, thus its usage significantly improves model quality as the user can make assertions and check if these are valid or not.

3.1. The use of OntoUML

OntoUML provides a well-founded UML profile. The classes in OntoUML are based on some important ontological meta-properties that allow the creation of consistent ontologies [Guizzardi 2005].

Examples of meta-properties are *identity principle* and *rigidity*. *Identity principle* is related to the nature of an object. For example, a Student is a Person, as they have the same identity principle, but they can never be a Car, as they have different identity principle. The *rigidity* principle is the capacity of an entity to be part of a class maintaining its existence. For example: John is an individual of the class Student but, in a given world, it can cease to be a Student and still exists as a Person. However, in any world John cannot cease to be a Person without ceasing to exist. Thus, Student is an example of an anti-rigid class while Person is an example of a rigid class.

Table 1 – OntoUML Class Stereotypes present in the Normative Acts Ontology

Stereotype	Main Characteristics	Example
Kind	Rigid types which provide an identity principle	Person, TV
Subkind	Relationally independent rigid specializations of kinds, collectives, or other subkinds	Man, LED TV
Category	Aggregate rigid elements with different identity principles	Animal, Electronic
Collective	Elements whose instances are collectives, i.e., they are collections of elements that have a uniform structure	A forest, a group of people

The OntoUML stereotypes present in the Normative Acts Ontology are summarized in Table 1 above. For an in depth presentation, formal characterization and empirical evidence for a number of the ontological categories underlying OntoUML, the reader is referred to [Guizzardi 2005].

4. The Domain Ontology for Brazilian Normative Acts

The domain ontology presented here is based on official documents: the Presidency Writing Manual (hereafter, for short, called PWM) and on the Brazilian Federal Constitution. Although the Normative Acts (NAs) Ontology created to the National Agency of Terrestrial Transportation involves three aspects of the Normative Process - Structural Elements, Management Issues and Regulatory Marks – the Reference Model presented here considers just the Structural Elements of NAs, considering NAs compositions, aggregations, its internal elements, and all relations between these.

The ontology reference model presented in detail in this section is divided in three subdomains and modeled using OntoUML diagrams. For highlight, the first occurrences of an ontology terms are presented with the Courier New font. It is important to mention that the ontology reference model is fully available for download at: http://www.nemo.inf.ufes.br/en/courses/ontologyengineering.

4.1. Normative Acts and Articles Subdomain

The model related to this subdomain states the different existing types of `Normative Acts` in Brazil and their internal structure.

Due to space limitations, the diagram that differentiates the NAs is not presented in image. This diagram's information was extracted from the 59th Brazilian Federal Constitution article, where it states that the legislative process involves the creation of: `Constitution Amendments`, `Complementary Laws`, `Ordinary Laws`, `Delegated Laws`, `Provisional Measures`, `Legislative Decrees` and `Resolutions`. `Decree`, `Ordinance` and `Handout` were extracted from the PWM. All of these NAs are disjoint from each other, i.e., no NA can be of two different types at the same time. NAs are defined by its composition by different subkinds of Articles and by its preliminary elements.

At the adopted conceptualization, the `Article` performs the most important function in NAs as it contains the statements defining the rules and information that the NA is about. In the ontology reference model every concept has a (direct or indirect) relation with Articles. It can be seen in Figure 2 that every article has an identifier number. An OCL rule formalizes that Articles in the same NA have unique natural numbers.

Figure 2 also represents the different types of Articles. Articles can be `Ordinary Articles` (regular Articles, the ones that states new communication), `Revocation Clauses` (Articles that revokes other Articles) or `Duration Clauses` (Articles that asserts a validation time). Every NA must be composed of at least one Ordinary Article, but it is not necessary for it to be composed of Revocation Clauses or Duration Clauses.

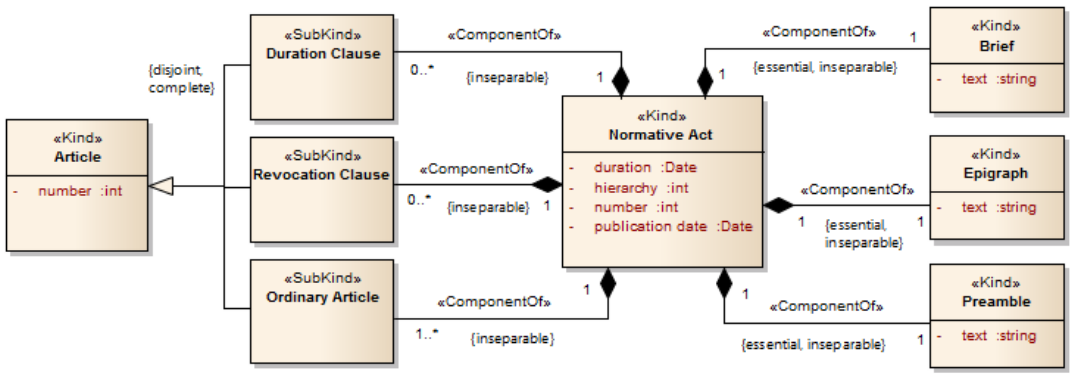


Figure 2. Compositions of Normative Acts

`Preamble`, `Epigraph` and `Brief` (in Portuguese: *Preâmbulo*, *Epígrafe*, and *Ementa*, respectively) are obligatory preliminary elements in every NA. These preliminary elements cannot be modified (vetoed, revoked or altered) – these properties are stated in the Ontology Reference Model by the composition meta-properties of essential and inseparable [Guizzardi 2005].

4.2. Discrimination of Normative Acts' Elements Subdomain

Grouping and Discrimination Elements are important part of NAs as they provide to their author the desired abstraction granularity. The `Discrimination Elements` are used to describe in more detail the information being normalized in a NA.

The different types of Discrimination Elements are: `Paragraph`, `Item`, `Letter` (in Portuguese: *Parágrafo*, *Inciso*, and *Alínea*, respectively) and `Letter Discriminator`. Every Discrimination Element is a part of a Normative Act because the Articles that are discriminated by these elements are part of the Normative Act.

Articles can be of two types: `Simple Article` or `Composed Article`. While the former consists only of a text, the latter consists of its introductory text, named `Caput`, and of at least one Item or Paragraph. Both types are represented in Figure 3.

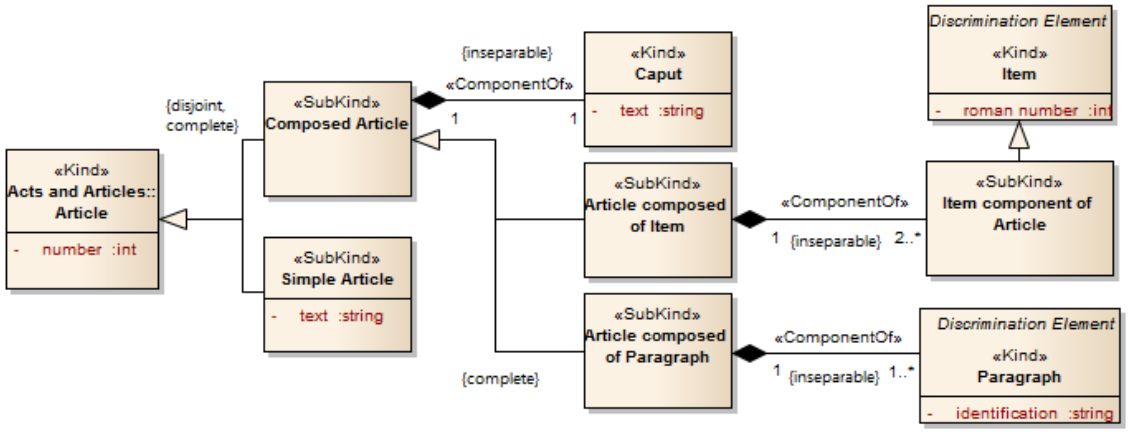


Figure 3. Compositions of Articles

If there is just one Paragraph composing an Article, its identification string must be *Unique Paragraph*. OCL rules ensure these restrictions.

Articles and Paragraphs can be decomposed in `Items` (see Figure 3), identified by roman numbers. Items, represented in Figure 4, can be `Simple Items` (when undivided) or composed by Letters. Letters can also be undivided (`Simple Letters`) or they can be divided in Letter Discriminators. The PWM states at its section 10.2.2.3 that "*Letters can be discriminated with cardinal numbers, followed by periods*". As no name is given to these discrimination elements, we named them Letter Discriminators.

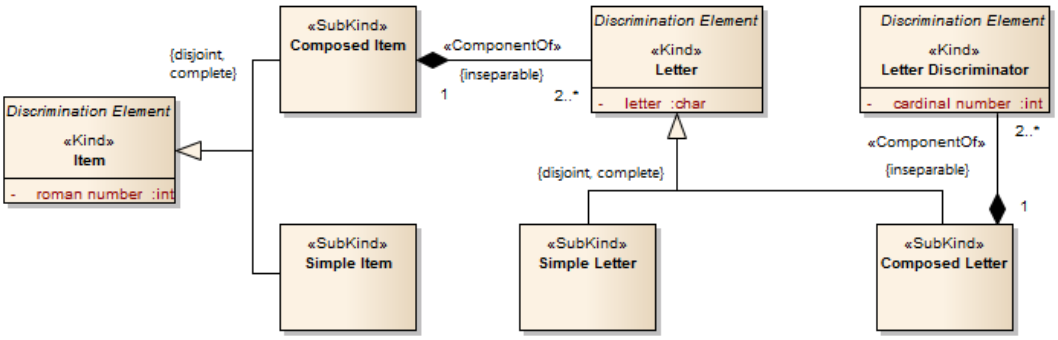


Figure 4 - Items, Letters and Letter Discriminators

4.3 Grouping of Normative Acts' Elements Subdomain

The `Grouping Elements` are used to easily aggregate related information in a NA. They are the `Parts`, `Books`, `Charters`, `Chapters` and `Sections` (in Portuguese: *Parte*, *Livro*, *Título*, *Capítulo*, and *Seção*, respectively). Every Grouping Element is part of the NA that is composed of the Articles that are grouped by these Grouping Elements.

As represented in Figure 5, related Articles can be grouped in Sections or Chapters. Sections can be of two types: `Simple Sections`, i.e., Sections that are not composed by other Sections, and `Composed Sections`, that are Sections composed by other Sections or `Subsections`: a specific type of Section that just occurs composing a Composed Section.

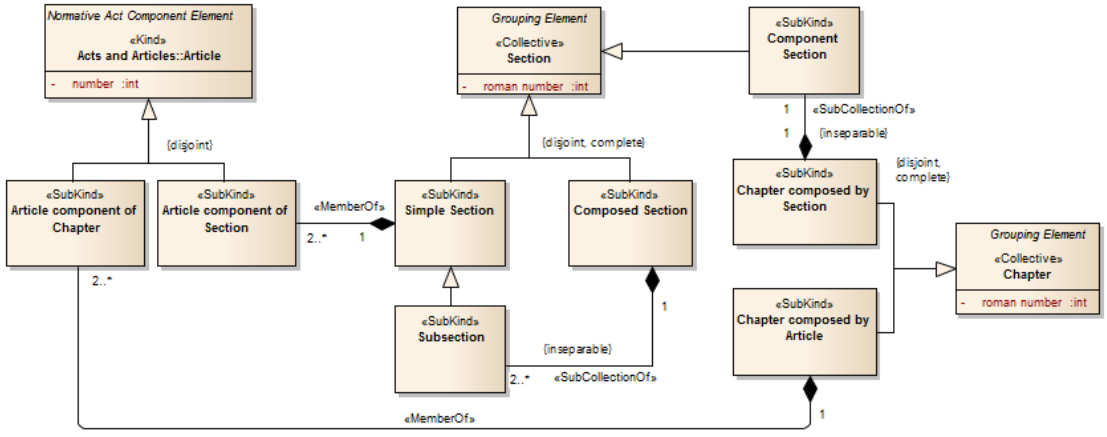


Figure 5. Grouping of Articles in Sections and Chapters

The grouping of Chapters by Charters, of Charters by Books, and of Books by Parts can be seen in Figure 6.

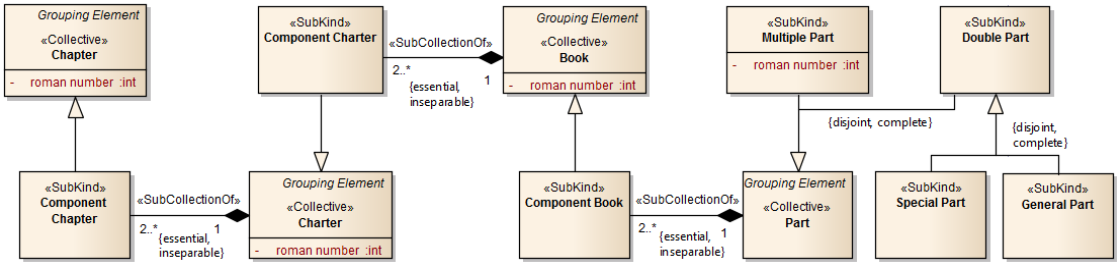


Figure 6 - Chapters, Charters, Books and Parts

Parts can be of types `Double Part` or `Multiple Part`. OCL rules were created to state that every NA that is composed by a Multiple Part always have at least three of these and to state that if the NA is composed of a Double Part it must have exactly only one instance of a `General Part` and a `Special Part`.

5. Practical Applications

As stated by [Guizzardi 2007], there is a clear distinction between (a) Conceptual Modeling, (b) Design and (c) Implementation. The ontology reference model presented here is a result of the Conceptual Modeling stage, as it aims to make a clear and precise description of the Normative Acts (NA) domain elements for the purposes of communication, learning and problem solving, independent of implementation platform or technology. Considering a Model Driven Architecture approach [Miller and Mukerji 2003], the Reference Model can be seen as the *Computational Independent Model* (CIM), its design products as *Platform Independent Models* (PIMs) and its implementation products as *Platform Specific Models* (PSMs). In the Design and Implementation phase, this conceptual specification is transformed by taking into consideration a number of issues ranging from architectural styles, non-functional quality criteria to be maximized, target implementation environment, etc. The same conceptual specification can potentially be used to produce a number of different artifacts in different implementation languages – ranging from relational databases to semantic web languages, like the Web Ontology Language (OWL).

In Brazil, according to the Federal Information Access Law (Brazilian Federal Law number 12,527, from November 18th 2011. Available just in Portuguese at http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm), every public organ or entity must use the Internet to publish, in detail, the formats used to structure their information. The ontology presented here, as a well-founded formalization of a domain conceptualization, is the best option to accomplish this requirement. This same law also states that the public organ or entity must enable automated access by external systems to the information in open, structured and machine-readable formats. Thus, an ontology implementation in the Resource Description Framework (RDF) or OWL can entirely satisfy the law. In fact, the federal government with the World Wide Web Consortium (W3C), cite OWL as a desirable practice for Open Data [Comitê Gestor da Internet no Brasil 2011].

A computational implementation of the proposed ontology reference model can also be used in applications to assist different stages of the regulatory and legislative process, like the NA edition. The usage of automation software can significantly improve the time spent with the huge amount of NAs that are created daily in a public entity. This automation can make processes more effective as it reduces the domain specialists' writing and research time and it gives them more time to think and reason about the subject of the documents.

In the edition phase, ontology-based software can eliminate from the writer the need to know each concepts from the PWM, as the software can automatically create these concepts for him. Another great improvement is the restriction to create documents that are not in accordance with the PWM. As an example, every time a writer creates an Article, if he wants to create a subtopic of this Article, he can just press the TAB key and two options appears to him: an Item or a Paragraph. So, it is not possible to create, for example, a Letter discriminating directly an Article. Another possible usage of this software would be to evaluate Normative Acts already created.

The ontology-based software for the NA process can be significantly improved by using other Ontology Models. The usage of an ontology model about NAs' revocation or modifications can add the ability to control the impact of a newly published NA in others already published. That is, there will be no need to manually state that an NA is repealed; it can be done automatically by software – thus, eliminating human errors.

Also, adding ontology reference models to the domain that the NAs are about, we can create a semantic annotation feature [Oren et al. 2006], where the writer can mark (tag) the concepts improving their semantics in accordance with the conceptual model. Please consider a reference model about highways, where a highway is composed by lanes and where it has a relation with other highways (for example "crosses"). If the NA editor is writing, for example, about BR-101 (a Brazilian highway), it can mark it as an instance of highway so, even if it is not explicit in the text that the BR-101 has lanes and that it can cross other highways, this information can be inferred. Semantic annotation is widely studied and used, even for large-scale scenarios [Dill et al. 2003].

With the usage of semantic annotation, the search on documents is improved, as there is no need to lexically match all search terms. A search using the word "highway" can return documents that use the term highway, and documents that deal with highways but do no explicitly use this concept, for example, a document that contains information about the BR-101. An improved search is vital to guarantee that new NAs do not conflicts with older NAs.

Already existent ontology-based tools for the legislative process can be found in [Valente and Breuker 1995] and [Gangemi et al. 2003]. The base ontologies used in these works, however, do not share the same scope of the ontology here presented and some of them are not formalized using a well-founded ontology language such as OntoUML.

6. Conclusions

This work presented an Ontology Reference Model for the domain of Normative Acts, built over information extracted from the Brazilian Constitution and the Presidency Writing Manual. The proposed ontology is formalized using an ontologically well-founded highly expressive language and it was developed following the ontology engineering methodology presented here. This ontology can be used as basis for automation and interoperation of information about this domain.

The Ontology Reference Model is presented considering three subdomains:

(a) Normative Acts and Articles, where different existing types of Brazilian Normative Acts are presented as well as their internal structure, representing their composition by articles and preliminary elements;

(b) Discrimination of Normative Acts' Elements, related to elements that are used to describe in detail the information being normalized in a Normative Act;

(c) Grouping of Normative Acts' Elements, that formalizes elements that are used to easily aggregate related information.

As a reference model, this ontology aims to make a clear and precise description of the domain elements for the purposes of communication, learning and problem solving. As an implementation, the model is able to solve problems related to the redaction and edition of Normative Acts, like reference finding, and cross-impact analysis. The ontology, in its conceptual or computational form, can be used to adequate public organs to the Federal Information Access Law, as every public organ or entity must use the Internet to publish, in detail, the formats used to structure their information.

Future work involves the expansion of the ontology to represent other aspects of Normative Acts, like temporal ones (e.g. an NA is valid from a beginning date and can have an end date). Another important future work is the creation of the computational ontology in semantic web languages, like OWL.

Acknowledgements. This research has been funded by the MGIC Project. We thank the ANTT employees and all project members who have somehow contributed to the development of this work. This research has also been funded by FAPES/CNPq (PRONEX 52272362/11).

References

Ajani, G., Boella, G., Lesmo, L., et al. (2009). Legal Taxonomy Syllabus version 2.0. In IDT.

Barcelos, P. P. F., Guizzardi, G., Garcia, A. S. and Monteiro, M. E. (may 2011). Ontological Evaluation of the ITU-T Recommendation G.805. In 2011 18th International Conference on Telecommunications. IEEE.

Boer, A., Van Engers, T. and Winkels, R. (2003). Using ontologies for comparing and harmonizing legislation. In Proceedings of the 9th international conference on Artificial intelligence and law - ICAIL '03. ACM Press.

Breuker, J. and Hoekstra, R. (2004). Core concepts of law: taking common sense seriously. In Proceedings of the Third International Conference (FOIS-2004). IOS Press.

Clara, B. L., Di Iorio, A. H. and Lerena, R. G. (2010). Ontologies , ICTs and Law The International Ontojuris Project. In Proceedings of LOAIT 2010 IV Workshop on Legal Ontologies and Artificial Intelligence Techniques.

Comitê Gestor da Internet no Brasil (2011). Manual dos Dados Abertos: Desenvolvedores.http://www.w3c.br/pub/Materiais/PublicacoesW3C/manual_dados_abertos_desenvolvedores_web.pdf.

Dill, S., Eiron, N., Gibson, D., et al. (2003). A Case for Automated Large Scale Semantic Annotation. Web Semantics: Science, Services and Agents on the World Wide Web 1.1, v. 1, n. 1, p. 115–132.

Engers, T. M. Van and Glassée, E. (2001). Facilitating the Legislation Process Using a Shared Conceptual Model. IEEE Intelligent Systems, v. 16, n. 1, p. 50–58.

Falbo, R. de A., Guizzardi, G. and Duarte, K. C. (2002). An Ontological Approach to Domain Engineering. Proceedings of the 14th international conference on Software engineering and knowledge engineering - SEKE '02, p. 351–358.

Gangemi, A., Prisco, A., Sagri, M.-T., Steve, G. and Tiscornia, D. (2003). Some ontological tools to support legal regulatory compliance, with a case study. In On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops.

Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. Enschede: Centre for Telematics and Information Technology University of Twente.

Guizzardi, G. (2007). On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. Proceedings of the 2007 conference on Databases and Information Systems IV: 7th International Baltic Conference, p. 18–39.

Jackson, D. (2002). Alloy: a lightweight object modelling notation. ACM Transactions on Software Engineering and Methodology (TOSEM), v. 11, n. 2, p. 256–290.

Lima, J. A. de O. (2010). Interoperabilidade Semântica no LexML. Panorama da Interoperabilidade no Brasil. Brasília: p. 74–79.

Miller, J. and Mukerji, J. (2003). MDA Guide Version 1.0.1. Object Management Group.

Oren, E., Möller, K. H., Scerri, S., Handschuh, S. and Sintek, M. (2006). What are Semantic Annotations?.

Sales, T. P., Barcelos, P. P. F. and Guizzardi, G. (2012). Identification of Semantic Anti-Patterns in Ontology-Driven Conceptual Modeling via Visual Simulation. 4th International Workshop on Ontology-Driven Information Systems (ODISE 2012).

Valente, A. and Breuker, J. (1995). ON-LINE: An architecture for modelling legal information. In Proceedings of the 5th international conference on Artificial intelligence and law. ACM.

Visser, P. and Bench-Capon, T. (1996). The Formal Specification of a Legal Ontology. In Proceedings of JURIX.

Integrating Tools for Supporting Software Project Time Management: An Ontology-based Approach

Glaice Kelly da Silva Quirino, Ricardo de Almeida Falbo

Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department – Federal University of Espírito Santo (UFES) – Vitória, ES – Brazil.

gksquirino@inf.ufes.br, falbo@inf.ufes.br

Abstract. *Project Management is a complex process involving several activities and a large volume of information. There are several tools offering partial solutions for supporting this process, increasing the need for integrating some of them, in order to provide a fuller support to the Project Management process. This paper presents an integration initiative aiming at semantically integrating dotProject, a web-based project management application, to ODE, an Ontology-based software Development Environment. This integration initiative focuses on the project time management, mainly for supporting the following activities: definition of project activities, allocation of human resource to these activities, and scheduling. This initiative was conducted partially following OBA-SI, an Ontology-Based Approach for Semantic Integration, and it was done using a domain ontology built from a Software Process Ontology Pattern Language.*

Keywords. *Project Management, Semantic Integration, Ontologies, Semantic Interoperability, Ontology Pattern Language.*

1. Introduction

Project Management is a process that aims at establishing and evolving project plans, defining activities, resources and responsibilities for the project, as well as providing information to assess the actual achievement and progress against the plans, in order to control the project execution [ISO/IEC 2008]. In order to support this process, several tools are needed, such as tools for project control, software process definition, resource allocation and scheduling. Ideally, these tools should work together, exchanging data and services. The use of several tools to support the same process without some degree of integration between them brings many problems such as rework and inconsistency.

ODE (Ontology-based Software Development Environment) [Falbo et al. 2005] is a process-centered Software Engineering Environment (SEE), which has been developed grounded on ontologies. ODE has several tools, some of them supporting the Project Management process, such as tools to support software process definition, resource allocation, estimation, and risk analysis. However, there are still project management activities that are not supported by ODE, such as project scheduling and tracking.

In order to increase the support offered by a SEE, two main approaches are typically used: (i) developing new tools already integrated to the SEE; (ii) integrating to

the SEE tools already available. In the first approach, the same group develops new tools as integrated pieces of the SEE, continuously expanding its functionality. In the second approach, the focus is on integrating tools produced by others. The main challenge in this case is to establish a common understanding of the meaning of the terms used by the tools, and to solve semantic conflicts between these tools and the SEE in which they should be integrated.

In the context of the ODE Project, the first approach was predominant in the first years of the project. This approach was in line with the main design premise originally established for the project, namely: if the tools in a SEE are built based on ontologies, integration can be more easily achieved, since the same set of ontologies is used for building different tools supporting related software engineering activities [Falbo et al. 2005]. However, more recently, we realized that adopting only this approach is not enough. Nowadays there are many free software tools available, and sometimes it is more appropriate to integrate an existing one to ODE, instead of developing a new tool. Moreover, this is especially important for allowing software organizations continue to use some tools to which they are already accustomed, preserving their organizational culture. Thus, we started to work also in integrating existing tools to ODE. The first effort in this direction was done to integrate Subversion (a version control system) to ODE [Calhau and Falbo 2010]. Since ODE is an ontology-based SEE, the integration approach adopted focus on the use of ontology as an interlingua to map concepts and services used by the different enterprise applications, in a scenario of access to data and services via a shared ontology, as pointed by Jasper and Uschold (1999).

Aligned to this new direction of the ODE Project, we decided to improve the support to Project Management in ODE by means of integrating a tool that could provide functionalities for scheduling software projects. Since the current version of ODE runs in the Web platform, we looked for a web-based free project management tool that provides such functionality, and, after comparing some of them, we decided to select dotProject[1]. dotProject is a free open source web-based project management system, which basically provides functionalities for managing tasks, schedules and communication with contacts. As pointed in the main page of dotProject[1], it does not provide all the functionalities required for managing projects.

However, each application (ODE and dotProject) runs independently and implements its own data and process models. These models are not shared between applications, leading to several conflicts, including technical, syntactical and, especially, semantic conflicts. As pointed by Izza (2009), this heterogeneity is considered one of the major difficulties in the integration problem. In this context, the adoption of an approach that helps reduce the complexity of this task is important.

In [Calhau and Falbo 2010], Calhau and Falbo developed OBA-SI (Ontology-Based Approach for Semantic Integration), an approach to semantic integration of systems that concentrates efforts on requirements analysis and conceptual modeling. In this approach, semantic integration is performed in a high abstraction level, promoting semantic agreement between the systems at the conceptual level. For this, ontologies are used to assign semantics to items shared between systems, proposing an integration

[1]http://docs.dotproject.net/index.php?title=Main_Page

process independent of technology and covering three layers of integration: data, services and process. Once OBA-SI is very aligned with the premises of ODE Project (using ontologies for building and integrating tools), we decided to adopt it in our integration initiative. However, since dotProject does not provide an API (Application Programming Interface) providing services, we decided to address integration only in the data layer.

In this paper, we present the semantic integration of dotProject to ODE, following OBA-SI. First, we defined the integration requirements, defining the integration scenario. Our focus is on integrating functionalities supporting the Project Time Management process, as defined in the PMBOK [PMI 2008], which involves the following activities: define project activities, sequence activities, estimate activity resources, estimate activity duration, and develop schedule. Second, we developed an ontology addressing this universe of discourse to be used to map the concepts and relations of both systems. This ontology, called Project Time Management Ontology (PTMO), was built by assembling patterns of the Software Process Ontology Pattern Language (SP-OPL) [Falbo et al. 2013]. Besides, we retrieved the structural conceptual models of the tools to be integrated. ODE's conceptual model was already available; dotProject's conceptual model, on the other hand, had to be excavated. With the ontology and the conceptual models of the tools to be integrated in hands, we established mappings between them, in order to assign semantics to the structural models of the tools. As a result, we achieved an integration model, which were used to design a mediator. Finally, we implemented the mediator application, integrating dotProject to ODE.

This paper is organized as follows. In Section 2, we present a brief review of the literature on topics relevant to the context of this work, namely: Project Management and Systems Integration. In Section 3, we present the PTMO ontology, discussing how it was built from SP-OPL. In Section 4, we present the semantic integration of dotProject to ODE, using OBA-SI. Section 5 discusses related works. Finally, in Section 6, we present the final considerations of this paper.

2. Project Management and Semantic Integration

According to the PMBOK [PMI 2008], "Project management is the application of knowledge, skills, tools and techniques to project activities in order to meet project requirements". It is a complex process that involves several sub-processes, among them the project planning is a major one. The PMBOK groups planning related activities in the Planning Process Group, which involves the processes performed to establish the project scope, define its goals, and develop the course of actions required to attain them. This group includes processes for Scope Management, Time Management, Cost Management, Quality Management, Human Resource Management, Communication Management, and Risk Management, among others.

As pointed in the introduction of this paper, ODE provides partial support for some of these processes, namely: Scope Management, Time Management, Quality Management, and Risk Management. Thus, we have worked to improve this support by means of developing new tools to ODE, or integrating existing tools to it. In this paper our focus is on the Project Time Management process, which includes five very inter-

related activities [PMI 2008]: define project activities, sequence activities, estimate activity resources, estimate activity duration, and develop schedule. ODE provides only partial support to this process, since it does not help in developing schedules. To improve this support, we decided to integrate dotProject to ODE.

However, when integrating these systems, conflicts arise. They were developed by different groups, in different points in time, and they have no concern with integration. Thus, they can be said heterogeneous, autonomous and distributed systems [Izza 2009]. Heterogeneous refers to the fact that each application implements its own data model defining its concepts in its own way. Autonomous means that each application runs independently of the other. Distributed means that they implement their data model in their own data repository and this repository is not shared with the other tool [Izza 2009].

In particular, there are semantic conflicts, and integration in the semantic level should take the intended meaning of the concepts in a data schema or in operation signatures into account [Izza 2009]. Basically, semantic conflicts occur because applications do not share a common conceptualization. In this context, ontologies can be used to deal with meaning and semantic heterogeneities. Ontologies can be used as an interlingua to map concepts and services used by the different systems, in a scenario of access to data and services via a shared ontology [Jasper and Uschold 1999]. Moreover, semantic integration should occur at the knowledge level [Park and Ram 2004], considering that applications must share the meaning of their terminologies.

Among the various approaches for integrating systems that consider semantics to integrate systems, there is OBA-SI (Ontology-Based Approach for Semantic Integration) [Calhau and Falbo 2010]. This approach considers that the integration process is analogous to the software development process, consisting of phases of requirements gathering and analysis, design, implementation, testing and deployment. OBA-SI focuses on the integration analysis phase, in which the semantics may be set. During this phase, semantic mappings are made between the conceptual models of the tools being integrated, using an ontology to assign meaning. This ontology should be a reference ontology, i.e., a solution-independent specification making a clear and precise description of the domain entities for the purposes of communication, learning and problem-solving [Guizzardi 2007].

The integration process of OBA-SI begins with integration requirements elicitation phase, when the goals and requirements must be established. In this phase, we need to define the activities of the business process to be supported, the systems to be integrated to support them, and the domain where the integration takes place. The output of this phase is the integration scenario. Once defined the integration scenario, integration analysis can be performed. Figure 1 shows the activities involved in this phase. First, the conceptual models of the tools to be integrated should be retrieved, as well as a reference ontology of the domain where the integration takes place should be selected or developed. Following, concepts and relations of the conceptual models of the tools to be integrated should be mapped to the concepts and relations of the ontology. These mappings are said vertical mappings. Once the structural models are semantically annotated, the integration model is developed. This model is mainly based on the domain ontology, but it can also include elements of the tools being integrated that do

not have a counterpart in the ontology. These elements, if present in both tools, should be directly mapped, by means of horizontal mappings [Calhau and Falbo 2010].

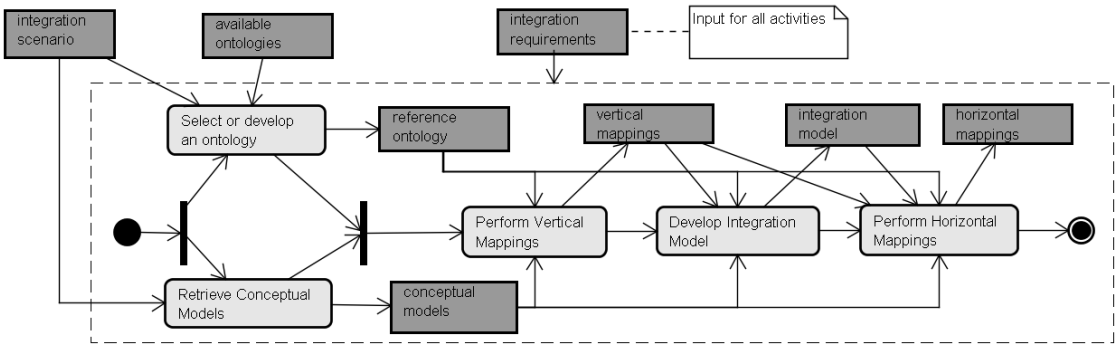


Figure 1. OBA-SI Analysis Phase

With the integration model in hand, an integration solution can be designed and implemented. There are several ways to design and implement an integration solution, and OBA-SI does not commit to any specific integration solution, although it proposes some guidelines for maintaining the semantic consistency in these steps.

In order to integrate dotProject to ODE adopting OBA-SI, we needed a reference domain ontology regarding project time management. Since several planning-related ODE's tools were developed based on the version of the Software Process Ontology presented in [Falbo and Bertollo 2009], we decided to use it. However, this ontology was reengineered in [Bringuente et al. 2011] to become aligned to the Unified Foundational Ontology [Guizzardi et al. 2008], and more recently it was defined as an Ontology Pattern Language (OPL)[2] [Falbo et al. 2013]. Thus, we decided to build a Project Time Management Ontology (PTMO) by assembling patterns of the Software Process Ontology Pattern Language (SP-OPL).

3. Using the Software Process Ontology Pattern Language to Develop a Project Time Management Ontology

SP-OPL is an OPL for the Software Process application domain. The main problem areas addressed by SP-OPL are Standard Software Process Definition, Project Process Definition and Scheduling, Resource Allocation, and Software Process Execution. In the next section, we discuss how we developed PTMO from SP-OPL.

SP-OPL has three entry points[3], depending on the focus of the ontology engineer. Considering our purposes, our entry point was the SPP (Software Process Planning) pattern, which considers the planning of the project process from scratch (i.e., without being based on a standard software process). The SPP pattern represents how a software process is planned in terms of sub-processes and activities, as well as it deals

[2] An OPL aims to provide holistic support for using Domain-related Ontology Patterns (DROPs) in ontology development for a specific application domain. It provides explicit guidance on what problems can arise in that domain, informs the order to address these problems, and suggests one or more patterns to solve each specific problem [Falbo et al. 2013].

[3] Each entry point allows the ontology engineer to focus on certain problems (and thus using certain patterns), disregarding others [Falbo et al. 2013].

with activity sequencing [Falbo et al. 2013]. Once defined the project activities, it is necessary to schedule the project and define the human roles required for performing the activities. To handle these aspects, the patterns PSCH (Process Scheduling) and HRP (Human Role Planning) were selected. The first defines the time window for project processes and activities, while the second defines the human roles responsible for performing a project activity [Falbo et al. 2013].

Human resource allocation was treated by reusing the PTD (Project Team Definition) and TDHRA (Team-dependent Human Resource Allocation) patterns. The PTD pattern regards the human resources that are member of a project team; the TDHRA pattern deals with allocating human resources to project activities, considering team allocation constraints. These patterns are in the Resource Allocation group of patterns [Falbo et al. 2013].

Finally, regarding process execution, we reused the PAET (Process and Activity Execution and Tracking) and HRPAT (Human Resource Participation and Tracking) patterns. The first registers the occurrences of processes and activities, taking into account the planned processes and activities, allowing to track the execution against to what was previously planned; the second registers the participation of human resources in activity occurrences, taking into account the existence of a prior allocation of these resources to planned activities [Falbo et al. 2013].

Figure 2 shows the conceptual model of the PTMO, resulting from the assembly of these patterns. This conceptual model is written in OntoUML, a UML profile that enables modelers to make finer-grained modeling distinctions between different types of classes and relations according to ontological distinctions put forth by the ontology of endurants of the Unified Foundational Ontology (UFO-A) [Guizzardi 2005]. Thus, the stereotypes shown in Figure 2 represent types of classes and relations as defined in UFO-A.

Project Processes are defined for a *Project*. There are two types of *Project Processes*: *General Project Process* and *Specific Project Process*. The first is the overall process defined for the *Project*. It consists of *Specific Project Processes*, thus allowing defining sub-processes. The second is composed by *Project Activities*, which may be, *Simple Project Activity* or *Composite Project Activity*. These activities are to be performed by human resources playing certain *Human Roles*. For example, a requirements specification activity defined for a project requires a requirements engineer to perform it. Once the project processes and activities are defined for a project, it is possible to establish the start and end dates for them, giving rise to *Scheduled Processes* and *Scheduled Activities*, respectively.

A *Human Resource Allocation* is the assignment of a *Scheduled Activity* to a *Human Resource* for playing a specific *Human Role*. A *Human Resource Allocation* depends on a *Project Team Allocation*, which allocates the *Human Resource* to the *Project Team* and indicates the role he/she will play in this team.

When scheduled processes and activities are executed, they generate *Process* and *Activity Occurrences*, respectively. Analogously to project processes, there are two types of processes occurrences: *General Process Occurrence*, which corresponds to the execution of the process as a whole, and the *Specific Process Occurrence*, which

corresponds to the execution of a particular project process. Similarly, there are two types of Activity Occurrences: *Simple Activity Occurrence*, which is an atomic action, and *Composite Activity Occurrence*, which is composed of other activity occurrences. Finally, when activities are performed (Activity Occurrence), they involve various *Human Resource Participations*, which refers to the participation of a single *Human Resource*.

Considering the *start* and *end dates* of *Scheduled Processes* and *Activities*, *Human Resource Allocations*, *Process* and *Activity Occurrences*, and *Human Resource Participations*, it is possible to track the project progress, contrasting what was planned (scheduled) with what actually happened (occurrences and participations).

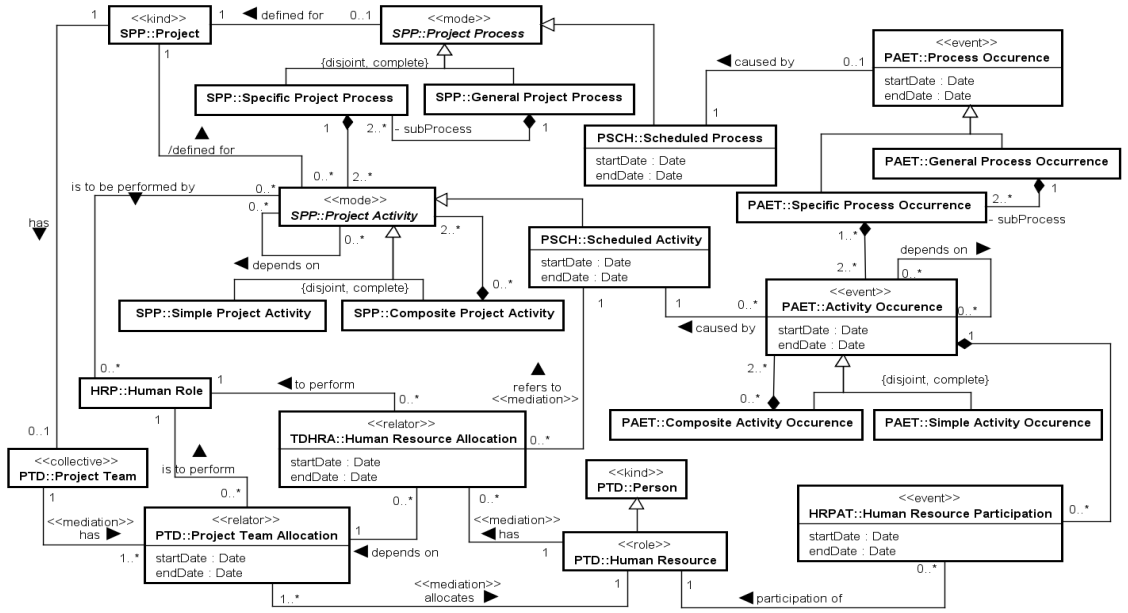


Figure 2. Project Time Management Ontology

4. Semantic Integration of` Software Project Management Tools

Once defined the integration scenario and the reference ontology, the required structural conceptual models had to be retrieved. Two different approaches were used. Since ODE was developed at NEMO, its structural conceptual model was available. On the other hand, the conceptual model of dotProject had to be excavated.

Figure 3 presents a fragment of ODE's structural conceptual model. It is presented only partially, due to space limitations. In ODE, a *General Project Process* is defined for a *Project*. This *General Project Process* is decomposed into *Specific Project Processes* that, in turn, are decomposed into *Activities*. During project activity definition, several process assets (resources, artifacts required and produced and so on) are defined for each activity, as well as sub-activities and dependencies between activities. All this information is registered in the *Activity Definition* class, which register also the scheduled start and end dates for the activity. For each activity, human resources can be allocated (*HRAllocation*), according to the demands informed during the process definition (*HRDemand*). When an activity is initialized, its actual start date is registered in the *Activity Execution* class, which represents the actual occurrence of

the previously planned activity. When a human resource spends some hours performing an activity to which she has been allocated, the *Expended Effort* must be registered.

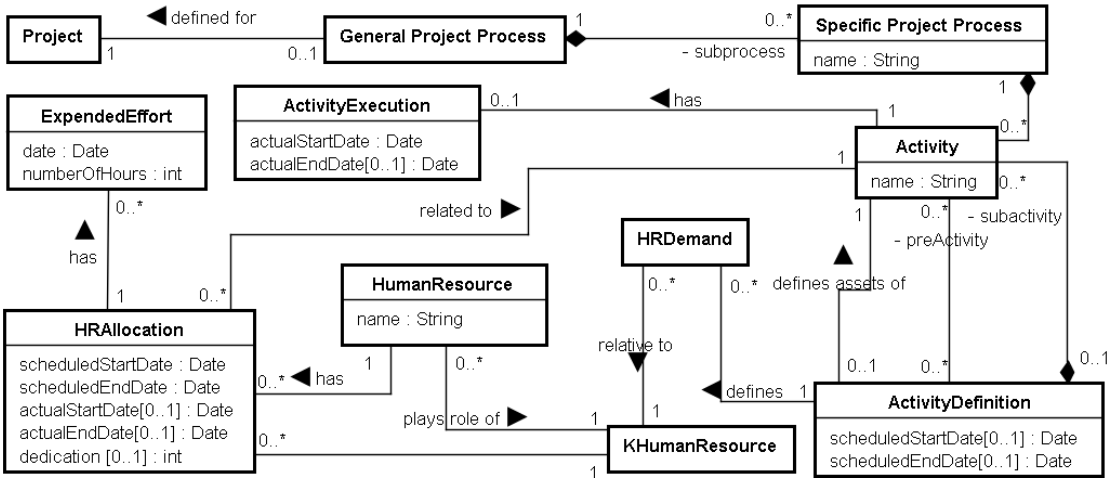


Figure 3. A fragment of ODE's Class Diagram

With respect to dotProject, we had to excavate its structural conceptual model. This was done by analyzing its database schema database and its graphical interface. Figure 4 shows a fragment of the structural conceptual model resulting from this step. As this figure shows, in dotProject, a *Project* has *Tasks*, to which *Contacts* can be allocated. *Tasks* can have sub-tasks and may depend on other tasks. Any events associated to a task can be registered by means of *Task Logs*.

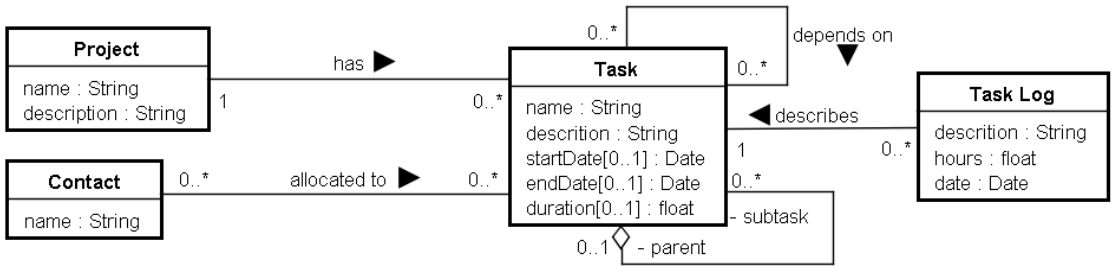


Figure 4. A fragment of dotProject's Class Diagram.

After retrieving the conceptual models of the tools, the next step is to assign semantics to their concepts and relationships by mapping them to concepts and relations of the reference domain ontology. These mappings, said vertical mappings [Calhau and Falbo 2010], allows comparing the concepts of the systems involved. Table 1 shows part of vertical mappings established to link the concepts of ODE and dotProject to the concepts of the PTMO ontology.

Project in ODE and dotProject are directly mapped to the concept of *Project* in PTMO, as well as *Human Resource* in ODE and *Contact* in dotProject that are directly mapped to the concept of *Human Resource* in PTMO. However, most of the concepts used in the tools are not directly mapped to a concept in PTMO. Contrariwise, in most cases, we need to consider attributes or relationships between classes to establish the same semantics of a concept in PTMO. For instance, the concept of *Simple Project Activity* in PTMO corresponds to a *Task* that does not have subtasks associated to it in

dotProject (Task.subactivity = null). In ODE, in turn, there are two concepts (*Activity* and *ActivityDefinition*) that map to the concept of *Project Activity* in PMTO. In order to know if a project activity is a simple or a composite project activity, it is necessary to see if the *ActivityDefinition* defines sub-activities for the corresponding *Activity*.

Table 1. Vertical Mapping of concepts

PTMO Ontology	ODE	dotProject
Project	Project	Project
Human Resource	Human Resource	Contact
Human Resource Allocation	HRAllocation	---
Project Activity	Activity + ActivityDefinition	Task
Simple Project Activity	Activity + ActivityDefinition, if ActivityDefinition.subactivity = null.	Task, if Task.subtask = null.
Composite Project Activity	Activity + ActivityDefinition, if ActivityDefinition.subactivity != null.	Task, if Task.subtask!=null
Scheduled Activity	Activity + ActivityDefinition, if (ActivityDefinition.scheduledStartDate != null and Activity.ActivityExecution = null).	Task, if (Task.startDate!=null and Task.startDate > currentDate)
Activity Occurrence	Activity + ActivityExecution	Task, if (Task.startDate != null and Task.startDate ≤ currentDate)

These types of mappings (direct and indirect) can also be observed in the case of vertical mappings between relationships, as shown in Table 2. The relationship "*Human Resource Allocation – refers to – Scheduled Activity*" in PTMO is directly mapped to the relationship "*HRAllocation – related to – Activity*" in ODE; whereas for mapping the whole-part relation between *Composite Project Activity* and *Project Activity* in PTMO to ODE, we need to cross two associations: "*ActivityDefinition – defines assets of – Activities*" and "*Activity – is sub-activity of – ActivityDefinition*".

Table 2. Vertical mapping of relationships

Ontology	ODE	dotProject
Human Resource Allocation – refers to – Scheduled Activity	HRAllocation – related to – Activity	Contact – allocated to – Task
Human Resource – has – Human Resource Allocation	Human Resource – has – HRAllocation	
Composite Project Activity – is composed by – Project Activity	ActivityDefinition – defines assets of – Activity; and Activity – is sub-activity of – ActivityDefiniton	Task – parent – Task

Once the structural models were semantically annotated, integration modeling started. In this step, first, the integration model was developed. The integration model is basically the conceptual model of the ontology plus some concepts arising from dotProject and others coming from ODE that do not have a counterpart in the ontology model. Due to space limitations, we do not present the integration model here.

Regarding the concepts added to the integration model, *Activity Occurrence Log*, for instance, was added to represent the dotProject's class *Task Log*, since, through *task logs*, it is possible to register the participations of human resources in activities (*Human Resource Participation* in PTMO).

With the integration model in hands, horizontal mappings were performed. In this step, the concepts that do not have a counterpart in the ontology model, and thus were introduced only in the integration model, were mapped. For instance, *Activity Occurence Log* in the integration model was mapped to *Task Log* in dotProject, and the relationship "describes" between *Activity Occurence Log* and *Activity Occurence* was mapped to the relationship "describes" between *Task Log* and *Task*.

Once established the horizontal mappings, the integration analysis phase is concluded, and we can start to design and implement the integration solution. For ODE and dotProject to communicate, it is necessary that the shared elements are translated. For doing that, we develop a mediator, which is responsible for translating data between the systems, as shown in Figure 5. The mediator is located inside ODE, making easier the access to ODE's database. In order to access dotProject's database, we implemented an interface for external communication, called dpClient[4], which behaves as an API to dotProject, since did not find any API available for dotProject that fits the purpose of our work.

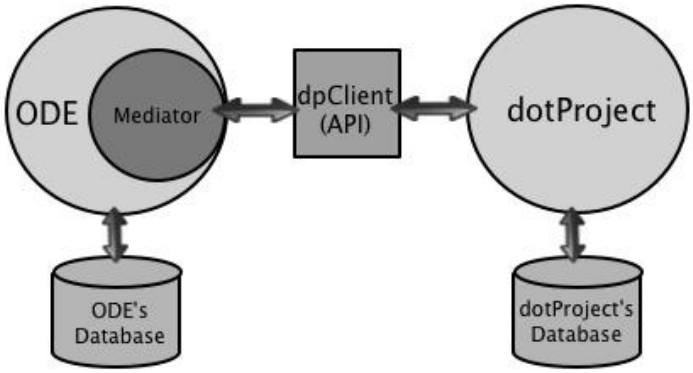


Figure 5. General Architecture of the solution for integrating dotProject to ODE

5. Related Works

Ontologies have been recognized as an important instrument for semantically integrating software applications [Izza 2009]. In the context of project management, Cheng et al. (2003) have used the Process Specification Language (PSL) Ontology for integrating Primavera P3, MS Project, Vite SimVision and 4D Viewer. Analogously to OBA-SI, the integration process used for building a distributed integration infrastructure also involves mappings between the concepts and relations of the involved systems and the concepts and relations of the PSL ontology. Moreover, there were also direct and indirect mappings, such as in our case (see examples given in the previous section). Although there are several similarities, there are also differences. The PSL Ontology deals with types of activities and activity occurrences. PTMO coverage, in turn, is

[4]https://github.com/glaice/dpclient

wider. It deals with the concepts of commitments (Project Process and Project Activity) and appointments (Scheduled Process and Scheduled Activities), in addition to the concept of occurrences (Process Occurrence and Activity Occurrence) as defined in UFO-C [Guizzardi et al. 2008]. Thus, it is possible to make finer distinctions, especially because the concepts of commitments and appointments are very important in project management. Regarding the technological solution for the integration, Cheng et al. develop wrappers for each application. The PSL wrappers are used to retrieve and transfer information between the applications, using PSL files. Not all scheduling and resource information is exchanged between the applications, since the granularity of the information may be different. Analogously, in our approach, the mediator is responsible for translating information from ODE to dotProject, using PTMO as an interlingua. However, in our case, changes made in dotProject do not reflect in ODE, since we have implemented the information exchange only from ODE to dotProject.

Concerning the methodological aspect, another work of semantic integration using OBA-SI is presented in [Calhau and Falbo 2010]. In their work, Calhau and Falbo integrated the version control system Subversion (SVN) to ODE. Access to SVN is worked by means of the svnkit library[5]. To translate data between the tools, a mediator stores information about the mappings between concepts and relationships of the tools being integrated and an ontology about the Software Configuration Management domain. Since in this work we also followed OBA-SI, the approach is quite similar.

6. Conclusions

This paper presented an initiative of semantically integrating dotProject, a web-based project management system, to ODE, an ontology-based Software Development Environment. This initiative was conducted partially following OBA-SI [Calhau and Falbo 2010], an Ontology-Based Approach for Semantic Integration, and it was done using a Project Time Management Ontology (PTMO), which was built from the Software Process Ontology Pattern Language [Falbo et al. 2013]. For implementing an integration solution, we developed a mediator responsible for exporting data from ODE to dotProject, allowing visualizing schedules in ODE, and thus providing a more complete support to the project management process in ODE.

We should highlight some limitations of our work. First, semantic integration is worked only in the data layer. Moreover, it occurs only from ODE to dotProject, i.e., data from ODE's database are passed to dotProject, but changes in dotProject's database are not reflected in ODE. Ideally, the integration should occur in both directions, and in other integration layers, especially in the service/message layer [Izza 2009]. Thus, there is room for adding new features to this work, or even integrating other tools in order to provide a wider support to the Project Management process.

Acknowledgments - This research is funded by the Brazilian Research Agencies FAPES/CNPq (PRONEX Grant 52272362/11).

[5] http://svnkit.com/

References

Bringuente, A. C. O., Falbo, R. A., Guizzardi, G. (2011), "Using a Foundational Ontology for Reengineering a Software Process Ontology". Journal of Information and Data Management, vol. 2, n. 3, pp. 511-526.

Calhau, R.F., Falbo, R.A. (2010), "An Ontology-Based Approach for Semantic Integration. Proceedings", Proc. 14th IEEE International Enterprise Distributed Object Computing Conference, Vitória, Brasil.

Cheng, J., Gruninger, M., Sriram, R. D., and Law, K. H., (2003), "Process Specification Language for Project Scheduling Information Exchange", International Journal of IT in Architecture, Engineering and Construction, vol. 1, n. 4, pp. 307 - 328.

Falbo, R. A., Ruy, F.B., Moro, R. (2005), "Using Ontologies to Add Semantics to a Software Engineering Environment". In: Proc. 17th International Conference on Software Engineering and Knowledge Engineering - SEKE'2005, Taipei, China.

Falbo, R. A., Bertollo, G. A (2009), "Software process ontology as a common vocabulary about software processes". International Journal of Business Process Integration and Management (IJBPIM), v. 4, p. 239-250.

Falbo, R. A., Barcellos, M.P., Nardi, J.C., and G. Guizzardi (2013), "Organizing Ontology Design Patterns as Ontology Pattern Languages," Proc. 10th Extended Semantic Web Conference, Montpellier, France.

Guizzardi, G. (2007), "On Ontology, Ontologies, Conceptualizations, Modeling Languages and (Meta) Models", In: Vasilecas, O., Edler, J., Caplinskas, A. (Org.). Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV, IOS Press, Amsterdam.

Guizzardi, G. (2005) Ontological Foundations for Structural Conceptual Models, University of Twente.

Guizzardi, G. Falbo, R.A. Guizzardi, R.S.S. (2008) "Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology", Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments, Recife, Brazil.

Izza, S. (2009) "Integration of industrial information systems from syntactic to semantic integration approaches", Enterprise Information Systems, Vol. 3, No. 1, February, pp. 1-57.

ISO/IEC (2008), ISO/IEC 12207: Systems and software engineering — Software life cycle processes, 2th edition.

Jasper, R., Uschold, M. (1999), "A Framework for Understanding and Classifying Ontology Applications", Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden.

Park, J.; Ram, S. (2004), "Information Systems Interoperability: What lies Beneath?", ACM Transactions on Information Systems, vol. 22, pg. 595-632.

PMI (2008), A Guide to the Project Management Body of Knowledge (PMBOK Guide), 4th edition, Project Management Institute, Inc.

Towards a Semantic Alignment of the ArchiMate Motivation Extension and the Goal-Question-Metric Approach

Victorio Albani de Carvalho[1,2], Julio Cesar Nardi[1,2], Maria das Graças da Silva Teixeira[2], Renata Guizzardi[2], Giancarlo Guizzardi[2]

[1]Research Group in Applied Informatics. Federal Institute of Espírito Santo, Campus Colatina, Colatina/ES, Brazil.

[2]Ontology and Conceptual Modeling Research Group (NEMO). Federal University of Espírito Santo, Vitória/ES, Brazil.

{victorio, julionardi}@ifes.edu.br, maria.teixeira@ufes.br, {rguizzardi,gguizzardi}@inf.ufes.br

Abstract. *Supporting Goal-Oriented Requirement Engineering (GORE) in a systematic and comprehensive way may require the combination of distinct goal-oriented approaches. However, due to lack of common semantics, to combine these approaches can be challenging. In this work, we propose a semantic alignment between two complementary goal-oriented approaches: the ArchiMate Motivation Extension and the Goal-Question-Metric. The approaches are semantically analyzed in light of the Unified Foundational Ontology (UFO), which provides real-world semantics for both languages, serving as a reference ontology to support the ontological analysis of concepts and relationships of both approaches and the alignment between them.*

1. Introduction

In the past decade, *Goal-Oriented Requirement Engineering* (GORE) became a de facto standard approach in *Requirements Engineering* (RE) [Kavakli and Loucopoulos 2005][Yu et al. 2011]. In this context, many GORE approaches have been proposed, but they use to focus on particular perspectives, and no single approach can address all the needs of this engineering process. However, these single approaches can be put to work together in order to compound a stronger and more complete GORE framework, which could benefit of the strengths of each approach [Kavakli 2002].

Considering this, the current paper focuses on the alignment of ArchiMate Motivation Extension and Goal-Question-Metric (GQM) approaches. ArchiMate is a modeling-based framework that has gained visibility both in academia and in industry. This framework provides a component called Motivation Extension (ME) that supports modeling of the enterprise's actual motivations or intentions by adopting the concept of "goal", among others [The Open Group 2012]. GQM [Basili, Caldiera and Rombach 1994], in turn, is an approach for evaluating the fulfillment of enterprise's goals. It is a well established and widely used approach [Boyd 2005] [Kaneko et al 2011]. By being complementary, ArchiMate ME and GQM can be adopted in tandem as a way to define a more comprehensive goal-oriented framework.

Combining different languages, however, clearly characterizes a *Semantic Interoperability* problem, since for combining languages one has to: (i) make clear the meaning of the primitives that compose the languages (i.e., the real-world semantics of the languages); (ii) establish the correct ontological relations between the alternative semantic domains [Guizzardi 2005].

One of the defining aspects of ontologies is their use in making explicit shared conceptualizations. *Reference ontologies* are kinds of ontologies used in an off-line manner to assist humans in tasks such as meaning negotiation and consensus establishment [Guizzardi 2005]. *Foundational* (or *top-level*) *ontologies* are domain-independent reference ontologies that describe very general concepts independently of a particular problem or domain (e.g., object, event, quality, action, etc.) [Guizzardi 2005][Guarino 1998]. These ontologies can provide real-world semantics for general representation languages and constrain the possible interpretations of their modeling primitives [Mika et al. 2004][Guizzardi 2005]. Also, by being adopted as a common reference, foundational ontologies can be used to map different representation languages and approaches [Cardoso et al. 2010].

In this paper, we propose the use of the Unified Foundational Ontology (UFO) [Guizzardi 2005] [Guizzardi et al. 2008] as a well-founded basis for defining (ontological) real-world semantics for the concepts of ArchiMate ME and GQM approaches. However, we have two basic semantic-based problems: (i) the ArchiMate ME and the GQM do not share the same semantics; and (ii) some of their concepts' semantics are not clearly defined. These problems, therefore, may lead different designers to assume distinct meanings and uses for the supposed same concepts. Thus, in order to properly use ArchiMate ME and GQM together, it is necessary to understand the semantics of the concepts of each approach, and how to map the concepts between the approaches. The choice for UFO as a foundational ontology here is motivated by a number of successful cases of using this ontology in the analysis, re-design and integration of different major modeling languages, including UML [Guizzardi 2005], *i** [Guizzardi et al. 2012] and ARIS [Cardoso et al. 2010], among others.

The remainder of this paper is organized as follows: Section 2 presents briefly ArchiMate ME, GQM and the used alignment process; Section 3 describes the fragment of UFO necessary for this work; Section 4 contains the proposed ontology-based alignment between ArchiMate ME and GQM; Section 5 describes a running example that illustrates how the semantic alignment may be put in practice; Section 6 presents some related works; and, finally, Section 7 draws final considerations.

2. Background: ArchiMate Motivation Extension (ME) and GQM

ArchiMate ME addresses the way the enterprise architecture is aligned to its context by using of motivational elements [The Open Group 2012]. It builds upon existing work on GORE, such as KAOS and *i** [Kavakli and Loucopoulos 2005], adopting interesting findings of these previous initiatives. The concepts defined by ArchiMate ME are *goal*, *stakeholder*, *driver*, *assessment*, *requirement*, *principle* and *constraint*.

To clarify the use of some concepts that are considered in this work, Figure 1 depicts a diagram developed by using of ArchiMate ME language. This diagram presents two *stakeholders*, each one having at least one *driver*, which can be shared. For each *driver*, a *goal* is defined. These *goals* are related to the *stakeholder* that commits

on pursuing it. Also, one can model that a fulfillment of a goal contributes positively (or negatively) to the realization of other goals.

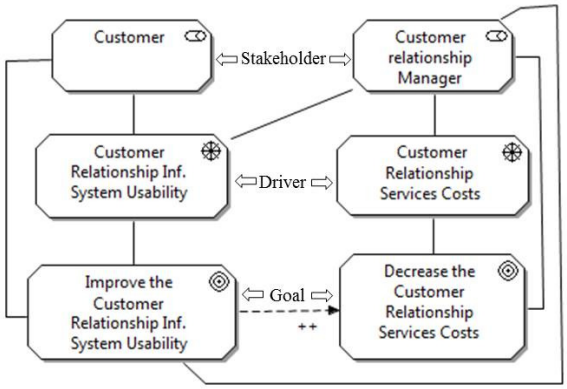


Figure 1. Illustrating some concepts of ArchiMate ME

GQM defines an approach for setting *goals* in a quality improvement paradigm. It is based on the assumption that for an organization to work efficiently with a measurement program, it should [Basili et al. 1994]: (i) specify the *goals* of the organization itself and its projects; (ii) map those *goals* to data that operationally define the *goals* (through *questions*, which direct the information that support the evaluation of a *goal*; and *metrics*, which indicate the types of data to be collected in order to answer the *questions*); and (iii) provide a framework for interpreting these data in relation to the established *goals*. Figure 2 presents an example of a GQM model for illustrating the relations between *goals*, *questions* and *metrics*.

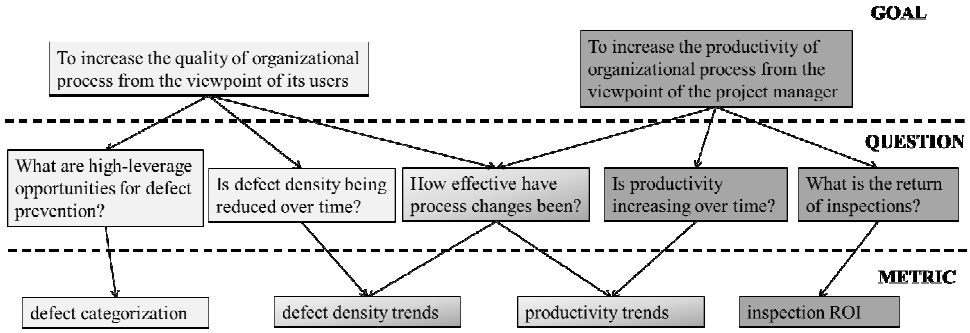


Figure 2. An example of the relations between *goals*, *questions* and *metrics* in GQM

In GQM, a *goal* is defined for an object (i.e. a *process*, a *product*, or a *resource*) based on a number of reasons (the *purpose*) with respect to models of quality (*issue* of quality) from *viewpoints* in relation to a particular environment. Thus, a *goal* consists of three coordinates (*object*, *issue/focus* and *viewpoint*) and a *purpose* [Basili et al. 1994].

Besides the fact that both approaches deal with *goals*, ArchiMate ME and GQM present some distinctions. While GQM focuses on measurement and evaluation of the fulfillment of organizational *goals*, ArchiMate ME focuses on specifying/representing these *goals*. In the context of GORE, a GQM model aims at describing *goals* that should have their fulfillment evaluated. ArchiMate, in turn, does not offer any approach for *goal* measurement. The ArchiMate ME supports the representation of organizational *goals* and their relations with other organizational elements. Some of these relations cannot be clearly expressed in GQM. Moreover, providing guidelines on how to identify

goals is out of the scope of ArchiMate ME. The aforementioned distinctions indicate that GQM and ArchiMate ME are complementary approaches. However, considering that the semantics of these approaches are not clearly defined and harmonized, a semantic alignment between them becomes necessary as a first step towards their use in tandem. For example, despite being a common term in both approaches, a deeper analysis reveals that the concept of *goal* in GQM specializes *goal* in ArchiMate ME, since the definition of *goal* in GQM comprises a set of interrelated concepts in ArchiMate (e.g., *goal*, *driver*, and *stakeholder*). We believe that understanding these semantic aspects may be useful for establishing an integrated framework.

For performing the ontological analysis and the alignment between ArchiMate ME and GQM we have followed the iterative process illustrated by Figure 3. This process is composed of three basic phases. In the "phase 1" and "phase 2", respectively, ArchiMate ME concepts and GQM concepts were analyzed and grounded in light of UFO. In "phase 3", such concepts were aligned taking as basis the ontological analyzes.

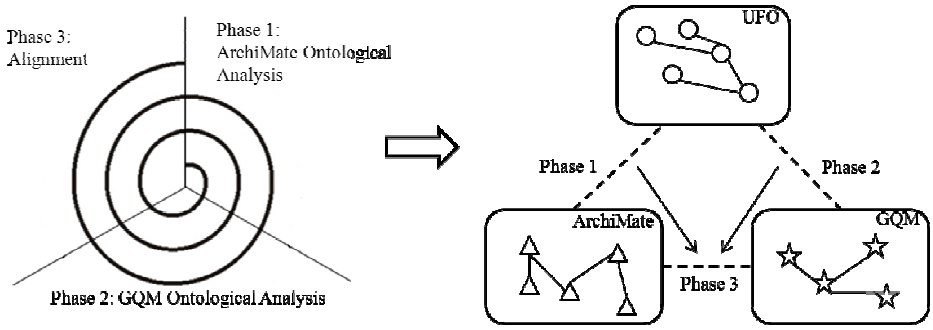


Figure 3. The iterative process used in the ontological alignment

3. The Unified Foundational Ontology (UFO)

UFO is a foundational ontology that has been developed with an interdisciplinary approach comprising theoretical and empirical results from Formal Ontology, Philosophical Logic, Linguistics, and Cognitive Psychology. UFO consists of three main parts: UFO-A, UFO-B, and UFO-C. UFO-A is an ontology of endurants [Guizzardi 2005], UFO-B is an ontology of events (perdurants) [Guizzardi et al. 2008][Guizzardi et al. 2013], and UFO-C is an ontology of social entities built on the top of UFO-A and UFO-B [Guizzardi et al. 2008].

A fundamental distinction in UFO is between individuals and universals. Universals are predicative terms that can be applied to a multitude of individuals, capturing the general aspects of such individuals. Individuals are entities that exist in reality possessing a unique identity and that can instantiate a multitude of universals [Guizzardi, 2005]. Figure 4 presents a fragment of UFO.

In UFO-A, endurants are individuals that are wholly present whenever they exist, and that can be further specialized into substantials and moments. Substantials are existentially independent endurants (e.g., a person, a car). Moments are individuals that can only exist in other individuals, being existentially dependent on their bearers (e.g., a person's headache, a covalent bond between atoms) [Guizzardi 2005]. Intrinsic moments are kinds of moments that are dependent on one single individual (e.g., John´s headache). Qualities are intrinsic moments associated with quality structures that inhere in an individual. A quality structure can be either a quality domain or a quality

dimension; quality domains are composed of multiple quality dimensions. For example, a color (quality) "c" of an apple (individual) "a" takes its value in a structure of three-dimensional color domain constituted of the quality dimensions "hue", "saturation" and "brightness". Modes are intrinsic moments that are not directly associated with a quality structure (and, hence, are not directly measurable) (e.g., John's intentions) [Guizzardi 2005]. Relators, in turn, are moments that existentially depend on two or more endurants [Guizzardi 2005]. For example, consider that John and Mary are married. In this case, the relator (their marriage) mediates the relation between John and Mary aggregating all externally-dependent modes that they acquire by virtue of participating in this relation (e.g, all commitments and claims towards each other). Thus, by virtue of being connected by this particular marriage, John plays the role of "husband" and bears responsibilities and rights towards Mary, who, by playing the role of "wife", and also bears the responsibilities and rights towards John.

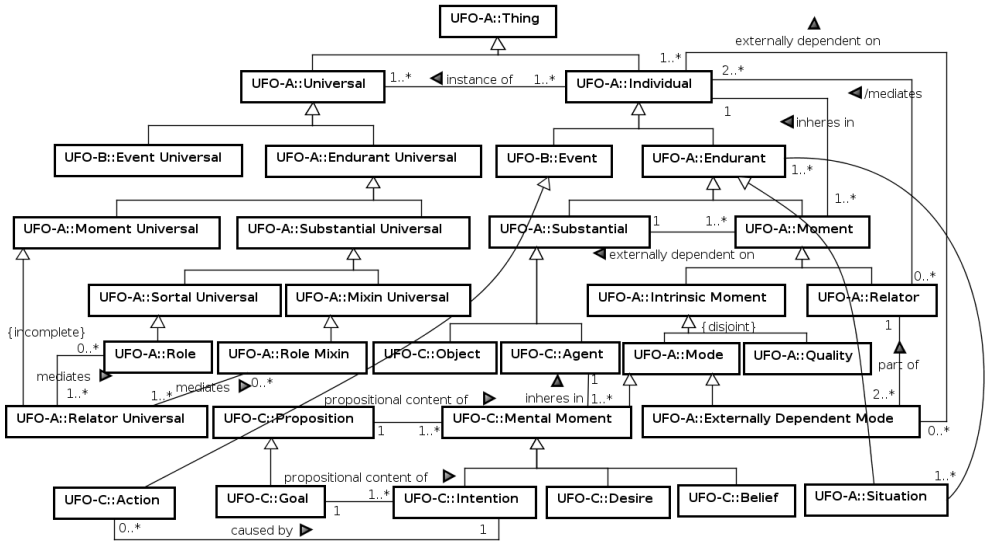


Figure 4. A fragment of UFO

Universals in UFO-A are types instantiated by endurants (universals) and can be substantial universal and moment universal, whose individuals are substantials and moments [Guizzardi 2005], respectively. Sortal universals are substantial universals that carry a principle of identity for their individuals (e.g., Apple, Person). The specialization of sortal universal is based on a meta-property called rigidity. An universal is rigid if it necessarily applies to its instances in every possible world (e.g., Apple, Person). In contrast to rigid universals, an universal is anti-rigid if it does not apply necessarily to all its instances. For example, an individual 'x', which is instance of the universal 'Student' in a world 'w_1' can cease to instantiate this universal in another world 'w_2' without ceasing to exist as the same individual. Roles are anti-rigid and relationally-dependent sortal universals (e.g., Student, Husband) [Guizzardi 2005], which means that roles are played by a substantial whenever there is a relator connecting it to other substantials. Role mixin represents an anti-rigid and externally-dependent non-sortal universal, which aggregates properties that are common to different roles (e.g., the role mixin 'Customer' aggregates properties from 'Personal Customer' and 'Corporate Customer') [Guizzardi 2005].

In UFO-B, events are individuals composed of temporal parts. They happen in time in the sense that they extend in time and accumulate temporal parts (e.g., a

conversation). Whenever an event is present, it is not the case that all its temporal parts are present. Events universals are patterns of features that can be realized in a number of different events [Guizzardi et al. 2008][Guizzardi et al. 2013].

In UFO-C, a basic distinction is the one between agents and (non-agentive) objects. Agents are agentive substantial individuals that are classified as physical agents (e.g., a person) or social agents (e.g., an organization). Objects are non-agentive substantial individuals that are classified as physical objects (e.g., a book) and social objects (e.g., a language) [Guizzardi et al. 2008]. Agents can bear special kinds of modes named mental moments. Mental moments refer to situations in reality (also called state-of-affairs, i.e., a portion of reality that can be comprehended as a whole) and has propositional content (an abstract representation of a class of situations referred by an intentional moment) [Guizzardi et al. 2008]. Mental moments are specialized in intentions, beliefs, and desires. Belief can be justified by situations in reality (e.g., my belief that the Moon orbits the Earth). Desires and intentions can be fulfilled or frustrated. Whilst a desire (e.g., a desire that Brazil wins the 2014 World Cup) expresses a will of an agent towards a state-of-affairs, intentions (e.g., go to the beach today) are desired state-of-affairs for which the agent commits to pursuing (i.e., intentions are self-commitments). Intentions cause the agent to perform actions and have propositional contents that is termed goal [Guizzardi et al. 2008].

4. Ontology-based Alignment between ArchiMate ME and GQM

This section presents the proposed semantic alignment between GQM and ArchiMate ME in light of UFO. Since *goal* is the common concept between these two approaches, it was considered as the key concept for performing the alignment.

In GQM, a *goal* is characterized as having a *purpose*, which is associated with three coordinates: *object*, *issue* and *viewpoint* [Basili et al. 1994]. According to GQM, an *object* can be a *process*, a *product*, or a *resource*. In terms of UFO, a *process* can be interpreted as an (complex) event [Guizzardi et al. 2008]. A *resource* is defined in UFO as a non-agentive substantial (i.e., an object) participating in an event [Guizzardi et al. 2008]. However, in organizational contexts the term "resource" is also used to refer to "human resource". In light of UFO, a human resource is a person (i.e., an agent) participating in an event playing a specific role. Therefore, the concept of *resource* in GQM comprises the idea of a non-agentive substantial (object) as well as the idea of an agent playing a role. In UFO, a *product* is a resource whose participation in events is limited to two types [Guizzardi et al. 2008]: creation participation (i.e., a *product* can be created) and changing participation (i.e., a *product* can be updated).

An *issue,* in GQM, refers to a quality aspect of an *object*, which can be interpreted based on the concept of quality in UFO [Guizzardi 2005]. Thus, an *issue* is as a quality that inheres in individuals (events and endurants). The individual that bears such quality (in terms of UFO) can be interpreted as being the *object* in GQM. For example, an *issue* can be the efficiency of a maintenance process (process as an *object* in terms of GQM) in an organization.

According to GQM, the fulfillment of a *goal* must be measured from a *viewpoint* (e.g., a manager's viewpoint, or a customer's viewpoint). Thus, the concept of *viewpoint* is associated with a "who" question, which makes reference to the roles played by one or more agents interested in a *goal*. Based on that, we may interpret that the *viewpoint*,

in terms of UFO, is associated with the role that an agent must be playing in the organizational context in order to be a possible measurer of the *goal's* fulfillment.

A *purpose* is related to a "why" question, in the sense that it refers to the intended effect associated with a *goal*. This intended effect is associated with a quality aspect (the *issue* coordinate) of an object (the *object* coordinate). For example, considering the *goal* "decrease the rate of error of the manufacturing process" the associated *purpose* is "decrease", the *issue* is "the rate of error", and the *object* is "manufacturing process". Thus, in light of UFO, we can interpret that the propositional content of the intention defines an intended effect (*purpose*) associated with a quality (*issue*) of an individual (*object*).

According to ArchiMate Specification 2.0, "a goal is defined as an end state that a stakeholder intends to achieve" [The Open Group 2012]. Thus, we can say that a *stakeholder* is committed (has an intention) to achieve a *goal*, and by achieving the *goal*, certain effects in reality are brought about. Thus, in terms of UFO, a *goal* (in ArchiMate ME) may be interpreted as "the propositional content of an agent's intention" [Azevedo et al. 2011]. In this context, we may state that GQM and ArchiMate ME can be aligned w.r.t. the concept of *goal*, given that in both approaches a *goal* may be interpreted, in terms of UFO, as a propositional content of an intention. As follows, however, we will discuss the concept of *goal* in GQM as a specialization of the concept of *goal* in ArchiMate ME.

As aforementioned, the GQM *viewpoint* coordinate can be interpreted as the roles played by agents that judge the fulfillment of (organizational) goals. These agents bear the intention of performing the evaluation of the (organizational) *goals*' fulfillment. On the other hand, there may be agents that bear the intention of pursuing the (organizational) *goals*' fulfillment, i.e., they are committed to perform actions in order to fulfill these *goals*. Thus, there may be agents that are committed at pursuing (organizational) *goals* and agents that are committed to judge if these *goals* were achieved or not. For example, in *goal* "Analyzing the customer relationship information system (*object*) for the *purpose* of improving its usability (*issue*) from the *viewpoint* of the customer" the "customers" judge the fulfillment of the *goal*, but they do not necessarily have a commitment at pursuing the *goal*. "Having a system with a great usability" may be only a desire for the "customers", whereas other *stakeholder*, possibly the "customer relationship manager", has the commitment at pursuing it. It is important to highlight, however, that the same agent may be committed to both, pursuing the *goals*' fulfillment and judging their fulfillment, although this is not desirable in organizational quality programs. This distinction between these two kinds of agents becomes clearer by the ontological analysis in light of the intentional concepts of UFO.

Interpreting the GQM's *goal* concept, we have realized that the three coordinates (*object*, *issue* and *viewpoint*) seem to characterize the problem being addressed by the *goal*, i.e., the "source" of the *goal*. Thus, in order to provide a semantic alignment between GQM and ArchiMate ME, the interpretation of the ArchiMate's *goal* concept is not enough. Besides that, it is also necessary to analyze two other ArchiMate ME's concepts used to model the "source" of the intentions: the *driver* and the *stakeholder.*

In Archimate ME, a *driver* is defined as "something that creates, motivates, and fuels the change in an organization". This definition is too vague and allows many interpretations. We consider that a *driver* may be interpreted as an event (external or

internal to the organization) that leads to a change of situation, which generates a *concern* about a key interest (e.g., *process*, *products*, *resources, costs, etc.*) of an organization. On the other hand, we can consider that some changes in organizations may also be motivated by some *stakeholder's concerns* despite explicitly considering the event that has generated the *concern*. In this case, a *driver* may also be interpreted as representing a *stakeholder´*s *concern*. According to [Azevedo et al. 2011], a *concern* is something that a *stakeholder* believes to be important. Therefore, a *concern* can be interpreted as "the propositional content of an agent's belief" [Azevedo et al. 2011]. The propositional content of the belief refers especially to properties believed to be important (the object of the *concern*) in a specific context. These two possible interpretations lead to a construct overload in Archimate ME language, since a *driver* may represent an event that generates a *concern* or the *concern* itself (without considering the event). In this context, it seems that the *driver* concept from ArchiMate ME is, somehow, related to the *object* and *issue* coordinates from GQM, in the sense that all of them refers to something a *stakeholder* is interested in, i.e., something the *stakeholder* believes to be important. In terms of UFO, the *driver* concept as well as the *object* and *issue* coordinates are related to the propositional content of a belief that refers especially to properties or characteristics believed to be important. For example, considering that the manager of a company has a *goal* of increasing the employees' productivity we can infer, in terms of UFO, that the manager (an agent) has a belief that the employees' productivity is important (the propositional content of the belief). Thus, in terms of ArchiMate, "the employees' productivity" may be represented as a *driver*. In terms of GQM "employees" may be seen as the value of the *object* coordinate (*a resource)* while the "productivity" refers to the *issue* coordinate. However, we remark that the *object* (as a *product*, a *process,* or as a *resource*) and *issue* coordinates together comprise only a subset of the possible *drivers* (in ArchiMate ME).

Finally, a *stakeholder* is defined by ArchiMate Specification 2.0 as "the role of an individual, team, or organization (or classes thereof) that represents their interests in, or concerns relative to, the outcome of the architecture" [The Open Group 2012]. According to UFO, a *stakeholder* can be interpreted as a role played by an agent (e.g., human individual, team or organization) able to refer to reality (in this case, "the enterprise architecture"). Thus, the agent instantiates a role, and, as consequence, the agent bears all the moments that characterizes that role, which include intrinsic moments (e.g., skills and capabilities that a person should have in order to play the role) as well as externally dependent modes associated with relators (e.g., as the rights and obligations that a person bears by participating on an employment contract). In this context, we may state that the concept of *stakeholder* in ArchiMate ME is aligned with the *viewpoint* coordinate of GQM since both can be interpreted, in light of UFO, as a role played by an agent. Table 1 summarizes the ontological analysis aforementioned.

Table 1. Summary of the ontological analysis of GQM and Archimate ME concepts

GQM	
Concept	Interpretation in light of UFO
Object	An *object* can be a *process*, a *resource*, or a *product*. *Process:* an (complex) event. *Resource:* a non-agentive substantial or an agent playing a role (human resource). *Product:* resource whose participation in events is limited to creation participation and changing participation
Issue	A quality that inheres in individuals (events and endurants)

Viewpoint	The roles played by agents that judge the fulfillment of (organizational) goals
Purpose	The intended effect expressed by the propositional content of an intention
Goal	The propositional content of an intention which defines an intended effect (*purpose*) associated with a quality (*issue*) of an individual (*object*).
Archimate ME	
Concept	Interpretation in light of UFO
Driver	An event that generates a *concern* or the *concern* itself. A *concern* can be interpreted as the propositional content of an agent's belief. The propositional content of the belief refers to the importance that the agent ascribes to something.
Stakeholder	A role played by an agent (e.g., human individual or organization) able to refer to reality.
Goal	The propositional content of an agent's intention

5. A Running Example of the Alignment between ArchiMate ME and GQM

This section presents an example that illustrates how the proposed semantic alignment between GQM and ArchiMate ME may allow an organization to use the strengths of each approach maintaining the traceability between the generated artifacts. In the example, ArchiMate ME is used for specifying the organizational *goals* and GQM for implementing an evaluation program regarding the fulfillment of these goals.

GQM establishes a structure for defining *goals.* By other hand, the ArchiMate ME does not define any specific structure for *goal*. Thus, we believe that the more detailed definition proposed by GQM can assist the designer in eliciting the organization's *goals*, i.e., the detailed structure provided by GQM can guide the designer in asking about and conceiving *goals*. Thus, we start this example by describing two organizational goals by using a GQM template, as follows: (G1) Analyzing the customer relationship process (*object*) for the *purpose* of decreasing its costs (*issue*) from the *viewpoint* of the CEO; and (G2) Analyzing the customer relationship information system (*object*) for the *purpose* of improving its usability (*issue*) from the *viewpoint* of the customer.

The proposed semantic alignment has shown that the concept of *goal* in GQM is a specialization of the concept of *goal* in Archimate ME. Thus, each *goal* described in GQM can be represented in an ArchiMate ME model. Also, it is possible to derive *drivers* in ArchiMate ME from *objects* and *issues* in GQM. For example, by the *goals* "G1" and "G2", we can infer, respectively, two *drivers* - (i) "customer relationship process costs", and (ii) "customer relationship information system usability" – which are represented with their respective *goals*, as Figure 5a.

Moreover, the alignment shows that it is possible to derive *stakeholders* in ArchiMate from *viewpoints* in GQM (i.e., the agents that judge the goal's fulfillment). For example, the "G1" *goal* is said to be measured by the *viewpoint* of the "CEO", which indicates that the "CEO" may be defined as a *stakeholder* in ArchiMate. Similarly, by "G2" *goal* we can derive another *stakeholder*: the "customer". Figure 5a represents these two *stakeholders*, associated with the respective *drivers*. Thus, an initial ArchiMate ME diagram (as presented by Figure 5a) is directly derived from *goals* described in GQM using the proposed alignment.

Moreover, although GQM addresses the agent committed to judge the *goals*' fulfillment, it does not offer a coordinate that is directly associated with the agent committed to pursue the *goal*'s fulfillment. ArchiMate Specification 2.0, in turn, offers

the *association* relationship as the only way to link *stakeholders* to *goals*. However, this relationship does not differentiate the *stakeholders* committed to pursue the *goals*' fulfillment and the *stakeholders* committed to judge it. By understanding these particularities, we suggest that the designer names each association for differentiating the "pursue" relations from the "judge" relations. For example, in Figure 5b, there are two *stakeholders associated with* the *goal* "improve the customer relationship information system usability": (i) the "customer", who is *associated with* that *goal* to represent the commitment at judging the goal's fulfillment; (ii) and the "CEO", who is *associated with* that *goal* for representing the commitment at pursuing the *goal*. The diagram in Figure 5b also presents the *stakeholder* "system analyst", who is committed at pursing the "Improve the Customer Relationship IS Usability" goal.

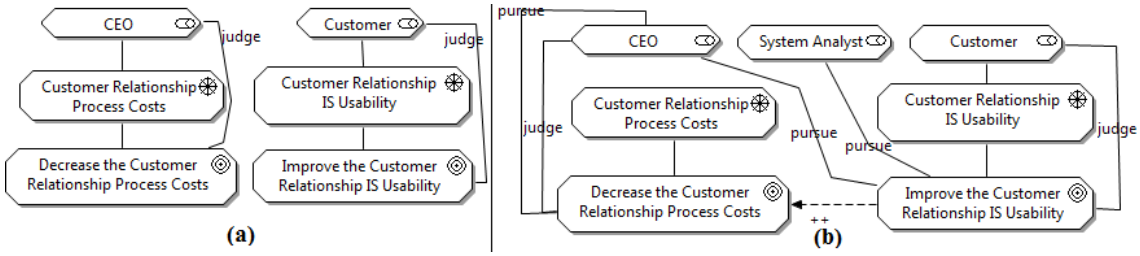


Figure 5 – An ArchiMate ME diagram derived from GQM (a) and an improved version (b)

As a result, the organization would have an ArchiMate ME diagram specifying the organizational *goals*, their sources and the relationships between them, aligned with a GQM model that could be carried out to evaluate the fulfillment of such *goals*.

6. Related Work

We are unaware of competing approaches which have attempted either an ontological analysis of GQM or semantic alignments between complementary goal-oriented design approaches. There are, nonetheless, in the literature a growing number of reports on the use (foundational) ontologies for performing analysis of (goal) modeling languages.

For instance, in [Cardoso et al. 2010], Cardoso and colleagues propose a semantic alignment between the ARIS framework (for business process modeling) and the TROPOS (for modeling and analysis of goals). Their proposal (which is also based on UFO) contributes to the establishment of a more comprehensive goal-oriented modeling approach by connecting a goal-modeling perspective with the modeling of business processes which are supposed to achieve these goals.

In [Azevedo et al. 2011], the authors perform a semantic analysis of the ArchiMate ME in light of UFO. This analysis was conducted by considering the whitepaper of the Motivation Extension. The ontological analysis performed in our work, however, has considered the ArchiMate ME Specification 2.0, which presents some differences of the initial version presented in the whitepaper, such as: (i) in the ArchiMate ME Specification 2.0 there is no longer the concept of "concern", which was replaced by the concept of "driver"; and (ii) the definition of the "role" concept in the standard suffered some changes, which were actually driven by the ontological analysis in [Azevedo et al. 2011]. So, in one sense, the analysis of ArchiMate ME performed here benefits directly from the previous work of Azevedo and colleagues. The two efforts, however, also differ also in focus: their work focuses exclusively on the real-

worlds semantics of this fragment of ArchiMate; our focus instead is in leveraging this semantics for integrating it with a complementary approach.

In [Soffer and Wand 2005], the authors employs some of the concepts of the BWW ontology to analyze the notion of goals in the context of Business Process Modeling (BPM). The account provided there is a language independent one and, in this sense, it is comparable to the analysis of hardgoals and softgoals present in UFO [Guizzardi et al. 2012]. In contrast with UFO, the view of goal provided by these authors take them as "sets of states of the domain". Under this view, goals are sets of elements closer to what is termed a situation (or a state of affairs) in UFO and, as such, are independent of intentions and, hence, independent of Agents (e.g., people, organizations). Such a view seems to fail to capture the requirements engineering and enterprise modeling intuition that goals are "desired state of affairs" [Yu et al. 2011]. Moreover, from an ontological standpoint, the UFO view of goals as *propositional contents of intentions which can possibly satisfied by sets of situations* allows even for goals which are unsatisfiable (an important analysis notion since the satisfiability of goals cannot always be defined a priori), as well as for distinct goals but which happens to be satisfied by exactly the same set of situations.

7. Final Considerations

Specification of organizational goals and the evaluation of the fulfillment of these goals are two complementary and essential activities. For supporting these activities in a systematic way, it may be necessary to combine distinct goal-oriented approaches. Due to lack of a common semantics between different approaches, their combination can be challenging. In this work, we propose a semantic alignment between the Archimate ME and GQM, which are two complementary approaches that can be applied in tandem to support the aforementioned activities. These approaches were aligned in light of UFO, which was used as a domain-independent reference ontology.

The ontology-based analysis conducted in this work contributed to clarify the meaning of some concepts of GQM and ArchiMate ME and to identify how these concepts can be aligned. For future works, we plan to continue the ontological analysis of the ArchiMate ME and the GQM, addressing other concepts, such as, the *assessment* concept. After that, we plan to define a cyclic process that comprises goal's specification and fulfillment evaluation, by adopting of ArchiMate ME (along with the GQM goal structure) to specify goals and GQM to assess the goals' satisfaction. We also plan to apply the proposed approach in real organizations in order to evaluate it.

Acknowledgments

This research is funded by the Brazilian Research Agencies CAPES/CNPq (402991/2012-5), FAPES/CNPq (PRONEX 52272362/11), FAPES (59971509/12) and CNPq (483383/2010-4, 310634/2011-3, and 311578/2011-0).

References

Azevedo, C., Almeida, J. P. A., Van Sinderen, M. J., Quartel, D. and Guizzzardi, G. (2011) "An Ontology-Based Semantics for the Motivation Extension to ArchiMate", In: 15th IEEE International Conference on Enterprise Computing (EDOC), Helsinki, Finland, p.25–34.

Basili, V. R., Caldiera, G. and Rombach, H. D. (1994) "The goal question metric approach". Encyclopedia of Software Engineering, v. 1, p. 528–532.

Boyd, A. J. (2005) "The evolution of goal-based information modelling: literature review". Aslib Proceedings - New Information perspectives, v. 57, n. 6, United Kingdon, Emerald Group Publishing Limited, p. 523–538.

Cardoso, E. C. S., Santos Jr., P. S., Almeida, J. P. A., Guizzardi, R. S. S. and Guizzardi, G. (2010) "Semantic Integration of Goal and Business Process Modeling", In: IFIP International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS), Natal/RN, Brazil.

Guarino, N. (1998) "Formal Ontology in Information Systems", In: Formal Ontology in Information Systems (FOIS), Trento, Italy, IOS Press, p. 3-15.

Guizzardi, G.(2005) "Ontological Foundations for Structural Conceptual Models". Centre for Telematics and Information Technology, University of Twente, The Netherlands.

Guizzardi, G., Falbo, R. A. and Guizzardi, R. S. S. (2008) "Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology", In: Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments. Recife, Brazil, p.244–251.

Guizzardi, G., Wagner, G. and Falbo, R. D. A. (2013) "Towards Ontological Foundations for the Conceptual Modeling of Events", In: 32nd International Conference on Conceptual Modeling (ER 2013), Hong Kong.

Guizzardi, R., Franch, X. and Guizzard, G. (2012) "Applying a Foundational Ontology to Analyze Means-end Links in the i* Framework", In: IEEE International Conference on Research Challenges in Information Sciences, p.1–11.

Kaneko, T., Katahira, M., Miyamoto, Y. and Kowalczyk, M. (2011) "Application of GQM+Strategies@ in the Japanese Space Industry", In: Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, v. 0, Los Alamitos, CA, USA, IEEE Computer Society, p. 221–226.

Kavakli, E. (2002) "Goal-Oriented Requirements Engineering: A Unifying Framework". Requirements Engineering, v. 6, n. 4, p. 237–251.

Kavakli, E. and Loucopoulos, P. (2005) "Goal Modelling in Requirements Engineering: Analysis and Critique of Current Methods", K. S. John Krogstie, Terry Halpin (Ed.); In: Information Modeling Methods and Methodologies: Advanced Topics in Database Research. p.102–124.

Mika, P., Oberle, D., Gangemi, A. and Sabou, M. (2004) "Foundations for Service Ontologies: Aligning OWL-S to DOLCE", In: The Thirteenth International World Wide Web Conference Proceedings. New Your, USA, p.563–572.

Soffer, P. and Wand, Y. (2005) "On the Notion of Softgoals in Business Process Modeling". Business Process Management Journal, v. 11, n. 6, p. 663–679.

The Open Group (2012). "Archimate 2.0 Specification". Berkshire, United Kingdon.

Yu, E. S. K., Giorgini, P., Maiden, N. and Mylopoulos, J. (EDS.) (2011) "Social Modeling for Requirements Engineering", Cambridge, Massachusetts, MIT Press.

Ontologies in Software Testing: A Systematic Literature Review

Érica F. Souza[1], Ricardo A. Falbo[2], N. L. Vijaykumar[1]

[1]Applied Computing – National Institute for Space Research (INPE)
São José dos Campos – São Paulo – SP – Brazil

[2]Department of Computer Science – Federal University of Espírito Santo (UFES)
Vitória – Espírito Santo – ES - Brazil

{erica.souza, vijay}@lac.inpe.br, falbo@inf.ufes.br

Abstract. *Ontologies have been widely recognized as an important instrument for supporting Knowledge Management (KM). In order to look for a domain ontology that can be used in KM in software testing, in this paper, we investigate, by means of a Systematic Literature Review (SLR), ontologies in the software testing domain, including questions related to their coverage of the software testing domain, and how they were developed.*

1. Introduction

Verification and Validation (V&V) activities intend to ensure that a software product is being built in conformance with its specification, and that it satisfies its intended use and the user needs [IEEE 2004]. V&V activities can be static or dynamic. Static V&V activities are typically done by means of technical reviews and inspections, and they do not require code execution; dynamic V&V activities involve code execution, and are done by means of testing [Mathur 2012]. Thus, Software Testing consists of dynamic V&V of the behavior of a program on a finite set of test cases, against the expected behavior [SWEBOK 2004].

Software testing processes generate a large volume of information. Thus, it is important to provide computerized support for tasks of acquiring, processing, analyzing and disseminating testing knowledge for reuse [Andrade et al. 2013]. In this context, testing knowledge should be captured and represented in an affordable and manageable way, and therefore, principles of Knowledge Management (KM) can be applied. Ontologies can be used for establishing a common conceptualization to be used in the KM system in order to facilitate communication, integration, search, storage and representation of knowledge [O´Leary 1998]. However, the software testing domain is very complex and building an ontology for it is not a trivial task. One of the main problems in the software testing literature is that there is not uniformity in the vocabulary used. In several cases, authors create and recreate concepts, using different terms. When analyzing different references, it becomes apparent that the terminology used is diverse.

Looking for a domain ontology that can be used in a KM initiative in software testing, in this paper, we investigate, by means of a Systematic Literature Review (SLR) [Kitchenham and Charters 2010], ontologies in the software testing domain. In this

SLR, we consider the following main research questions: (i) What is the coverage of the software testing domain in the existing testing ontologies? (ii) How were these ontologies developed?

This paper is organized as follows. Section 2 discusses briefly the background for this paper. Section 3 presents the SLR we performed. Section 4 discusses important findings identified during data analysis. Finally, Section 5 presents our conclusions.

2. Background

The testing process consists of several activities, namely: Test Planning, Test Case Design, Test Execution and Test Result Analysis. Like several other aspects of a project, testing must be planned. Test planning should be documented in a Test Plan. Test Case Design aims at designing the test cases to be run. Test Cases should be documented, and implemented as Test Scripts. During test execution, test cases are run, producing actual results, which should also be documented. Finally, in the Test Result Analysis phase, test results are evaluated to determine whether or not tests have been successful. Testing techniques and test criteria are used to support designing test cases. Moreover, testing usually is performed at different levels. Three important test levels can be distinguished, namely: Unit Testing, Integration Testing and System Testing [SWEBOK 2004, Mathur 2012].

During the testing process, a significant volume of information is generated. Such information may turn into useful knowledge to potentially benefit future projects from experiences gained from past projects [Andrade et al. 2013]. However, converting this information into applicable knowledge is not an easy task. There is a need to properly represent and process the knowledge so that it can be manageable. In this context, principles of Knowledge Management (KM) can be applied. Ontologies are a key technology for KM. They provide a shared and common understanding of a domain that can be communicated between people and application systems. Their use offers an opportunity for improving KM capabilities in large organizations [Davies et al. 2003]. In ontology-based KM systems, ontologies are mainly used for the following three general purposes [Abecker and Elst 2009]: (i) to support knowledge search, retrieval, and personalization; (ii) to serve as basis for knowledge gathering, integration, and organization; and (iii) to support knowledge visualization.

In order to find a domain ontology to be used as basis for a KM initiative in software testing, we conducted a Systematic Literature Review (SLR) aiming at inspecting the existing software testing ontologies. The research method applied was defined based on the guidelines for SLRs given by Kitchenham and Charters (2010). According to them, a SLR is a form of secondary study that uses a well-defined method to identify, analyze and interpret the available evidences in a way that is unbiased and (to a degree) repeatable. A secondary study is a study that reviews primary studies related to specific research questions with the aim of integrating/synthesizing the evidences related to these questions. A SLR involves three main phases: (i) Planning: refers to the pre-review activities, and aims at establishing a review protocol defining the research questions, inclusion and exclusion criteria, sources of studies, search string and mapping procedures; (ii) Conducting: regards searching and selecting the studies, in

order to extract and synthesize data from them; and (iii) Reporting: is the final phase and aims at writing up the results and circulating them to potentially interested parties.

3. The Systematic Literature Review

This section presents the Systematic Literature Review (SLR) we perform to investigate existing ontologies in the software testing domain. In Subsection 3.1, we present the main parts of the review protocol. In Subsection 3.2, we briefly describe the selected studies. Finally, in Subsection 3.3, we synthesize data extracted from the studies.

3.1. Review Protocol

Research Questions: This SLR aims at answering the following research questions:

RQ1. What is the coverage of the software testing domain in the existing ontologies about this domain?

RQ2. How were they developed?

Inclusion and Exclusion Criteria: The selection criteria are organized in one inclusion criterion (IC) and five exclusion criteria (EC). The inclusion criterion is: (IC1) The study presents an ontology about the software testing domain. The exclusion criteria are: (EC1) The study does not have an abstract; (EC2) The study is just published as an abstract; (EC3) The study is not written in English; (EC4) The study is an older version (outdated) of another study already considered; and (EC5) The study is not a primary study, such as editorials, summaries of keynotes, workshops, and tutorials.

Sources: Search was done in eight electronic databases that were considered the most relevant according to [Dyba et al. 2007], namely: IEEE Xplore, ACM Digital Library, SpringerLink, Scopus, ISI of Knowledge, DBLP Computer Science Bibliography, Science Direct, and Compendex.

Search String: The search string is the following: *("Software Testing" OR "Software Test") AND ("Ontology" OR "Ontologies")*. It was applied in three metadata fields (title, abstract and keywords). The search went through syntactic adaptations according to particularities of each source.

Assessment: Before conducting the SLR, we tested the review protocol. This test was conducted in order to verify its feasibility and adequacy, based on a pre-selected set of studies considered relevant to our investigation. The review process was conducted by one of the authors and the other two carried out its validation. They analyzed approximately 35% of the studies using two different samples.

3.2. Selected Studies

Using the search string, 396 records were retrieved. The selection process applied on the returned publications was performed in three stages. In the first stage, duplicates were eliminated by examining title and abstract, since several publications are available in more than one source. In the second stage, inclusion and exclusion criteria were applied considering also title and abstract. Finally, in the third stage, the exclusion criteria were applied considering the entire text. After applying the selection criteria, 18 studies remained. Table 1 shows the progressive reduction of the number of studies throughout the selection process for the review.

Table 1. Result of the Selection Process Stages of the SLR

Stage	Criteria	Analyzed Content	Initial Studies	Final Studies	Reduction (%)
1st	Eliminating duplication	Title and abstract	396	295	25.5%
2nd	IC1, EC1, EC2, EC3, EC4 e EC5	Title and abstract	295	30	89.8%
3rd	IC1, EC4, EC5 e EC6	Entire Text	30	18	40%

From the 18 studies, 12 different ontologies were identified. This difference comes from the fact that some papers present different parts or evolutions of the same ontology. As the result of this SLR, we ended up in the following testing ontologies: STOWS (Software Testing Ontology for Web Service) [Huo et al. 2003, Zhu and Huo 2005, Hong 2006, Yufeng and Hong 2008, Zhu and Zhang 2012], OntoTest [Barbosa et al. 2006, Nakagawa et al. 2009], TaaS Ontology [Yu et al. 2008, Yu et al. 2009], and the ontologies proposed in [Li and Zhang 2012], [Arnicans et al. 2013], [Guo et al. 2011], [Nasser et al. 2009], [Bai et al. 2008], [Ryu et al. 2011], [Sapna and Mohanty 2011], [Cai et al. 2009] and [Anandaraj et al. 2011]. From the 12 identified ontologies, we analyzed whether there were extensions, evolutions and/or other publications that present the ontologies more completely. It was the case of OntoTest. OntoTest has a testing resource sub-ontology presented in [Barbosa et al. 2008]. However, this study did not return in the SRL, probably because the searched sources do not contain this paper or because they failed to identify it by the search string.

3.3. Data Synthesis

After selecting the primary studies, we analyzed each one in order to answer the research questions presented in Subsection 3.1. Next, we present the data synthesis regarding these questions.

RQ1. *What is the coverage of the software testing domain in the existing ontologies about this domain?*

Regarding domain coverage, we notice that most of the ontologies have very limited coverage. The ontology presented in [Guo et al. 2011] specifies only the concept of test case. The one in [Li and Zhang 2012] focuses also on test case, but considering some concepts related to test process. Bai et al. (2008) presented an ontology, called Test Ontology Model (TOM), to model only testing artifacts and relationships between them. The ontologies presented in [Arnicans et al 2013], [Cai et al. 2009] and [Anandaraj et al. 2011] are, in fact, taxonomies. These ontologies only present a simple structure of the domain concepts of software testing, and thus they do not qualify as ontologies, or, at most, they are lightweight ontologies.

The ontology presented in [Nasser et al. 2009] is devoted to state machine based testing. The ontology presented in [Sapna and Mohanty 2011] focuses on scenario-based testing, though it captures general testing concepts too. The one presented in [Ryu et al. 2011] is not properly a testing ontology, but it is an OWL implementation of a specific testing maturity model developed by the authors (the Ministry of National Defense-Testing Maturity Model (MND-TMM)).

The ontologies that have higher coverage are: STOWS (Software Testing Ontology for Web Service) [Huo et al. 2003, Zhu and Huo 2005, Hong 2006, Yufeng and Hong 2008, Zhu and Zhang 2012], OntoTest [Barbosa et al. 2006, Barbosa et al. 2008, Nakagawa et al. 2009], TaaS Ontology [Yu et al. 2008, Yu et al. 2009].

STOWS classifies its concepts into three categories: (i) elementary concepts, which are general concepts about computer software and hardware; (ii) basic testing concepts, which include the concepts of Tester, Artifact, Activity, Context, Method, and Environment; and (iii) compound testing concepts, which combine basic testing concepts, giving rise to the concepts of Task and Capability. STOWS presents a set of taxonomies of each basic testing concept, including also some properties and few relations.

The TaaS Ontology has two core concepts (Test Task and Test Capability), which are composite concepts aggregating other concepts. Test Task consists of Test Activity, Test Type, Target Under Test, Test Environment, and Test Schedule. Test Capability, in turn, consists of Test Type, Test Activity, Test Environment, Target Under Test and Quality of Service.

Finally, OntoTest is a modular ontology, built in layers. OntoTest is composed of a "Main Software Testing Ontology", and six sub-ontologies [Barbosa et al. 2006]: Testing Process, Testing Phase, Testing Artifact, Testing Step, Testing Resource, and Testing Procedure sub-ontologies. The Main Software Testing Ontology is presented in [Barbosa et al. 2006]. It is a simple model that includes six concepts. According to this model, a Testing Process is composed of Testing Steps, and it has Testing Phases. A Testing Step requires Testing Resources, adopts Testing Procedures, consumes and generates Testing Artifacts, and depends on other Testing Steps. Testing Artifacts can depend on other Testing Artifacts, and can be composed of other Testing Artifacts. Finally, a Testing Procedure can be supported by Testing Resources, and is adequate to Testing Process. OntoTest Testing Step sub-ontology introduces the concept of Testing Activity, indicating that a Testing Step is composed of Testing Activities, while Testing Activities are not further decomposed. The remainder of this sub-ontology consists of two large taxonomies: a Testing Step taxonomy, and a Testing Activity taxonomy. The Testing Resource sub-ontology [Barbosa et al. 2008] has a taxonomy of types of resources. This taxonomy is organized in two branches: Human Resources (which can be members of Test Teams), and Testing Environment, which is further extended in Software and Hardware Resource. Software Resource is further extended into Testing Tool and Supporting System. Testing Tool can be composed of several types of Testing Modules. We did not find papers presenting the Testing Process, Testing Phase, Testing Artifact, and Testing Procedure sub-ontologies. So, we suppose that OntoTest is a work in progress.

RQ2. *How were the testing ontologies developed?*

With respect to this research question, we focused on some aspects related to the way the ontologies were engineered, namely: (i) Do the ontologies try to capture a common (shared) conceptualization of the testing domain, taking into account different references and especially international standards? (ii) Are the ontologies developed following an ontology engineering method (including some sort of evaluation)? (iii) In which abstraction level (conceptual and implementation levels) are the ontologies

developed? Which are the languages used? (iv) Do the ontologies take foundational aspects (foundational ontologies) into account?

The first aspect investigated is if the ontologies try to capture a common conceptualization of the testing domain. Some ontologies take international standards into account: OntoTest is based on 1st edition of ISO/IEC 12207; the ontology presented in [Arnicans et al. 2013] was created based on the glossary "Standard glossary of terms used in Software Testing" of the International Software Testing Qualifications Board – ISTQB; the ontology presented in [Bai et al. 2008] is based on the Unified Modeling Language 2.0 Test Profile (U2TP); and the ontologies presented in [Cai et al. 2009] and [Sapna and Mohanty 2011] are based on the SWEBOK [SWEBOK 2004]. The other studies neither mention the use of international standards as basis for their ontologies, nor which references were used as basis for developing the ontologies. The exception is the ontology presented in [Ryu et al. 2011], which, as said before, is an OWL implementation of the Ministry of National Defense-Testing Maturity Model (MND-TMM). It is worthwhile to point out that, despite some ontologies are based on international standards, generally they take only one standard into account, and thus they do not consider a broad set of testing references to really establish a common (consensual) conceptualization.

Regarding the methods adopted for building the ontologies, Arnicans et al. (2013) propose a method for semi-automatic obtaining lightweight ontologies, which uses the ONTO6 method. In [Sapna and Mohanty 2011], ideas were adapted from two methods for building ontologies: METHONTOLOGY [Juristo et al. 2007] and Ontology Development 101 [Noy and McGuinness 2001]. Cai et al. (2009) used the Uschold and King's skeletal method [Uschold and King 1995] for building their testing ontology. Finally, OntoTest was built using a method that combines guidelines given by SABiO [Falbo et al. 1998] and METHONTOLOGY, with focus on ontology capture and formalization. Finally, Anandaraj et al. (2011) followed a very simple method, comprising four steps, namely: (i) determine domain and scope of the ontology; (ii) define concepts in the ontology; (iii) create a class hierarchy; and (iv) define properties and constraints. The other studies do not mention if a method (or which method) was used for building the proposed ontologies.

Although the aforementioned ontologies have been developed following methods that include activities devoted to ontology evaluation, such as Uschold and King's skeletal method, SABiO and METHONTOLOGY, none of the studies discusses how the ontologies were evaluated, except [Arnicans et al. 2013], which says that a software testing expert has analyzed the ontology fragment related to testing techniques.

Regarding the abstraction level, 7 of the 12 studies (58.3%) present their ontologies as conceptual models, namely: STOWS, OntoTest, TaaS Ontology and the ontologies presented in [Li and Zhang 2012], [Arnicans et al. 2013], [Bai et al. 2008] and [Sapna and Mohanty 2011]. 5 of the 12 studies (41.7%) present the ontologies only as a code artifact (implemented in OWL), namely: the ontologies presented in [Guo et al. 2011], [Nasser et al. 2009], [Ryu et al. 2011], [Cai et al. 2009] and [Anandaraj et al. 2011]. The following ontologies are represented in both conceptual and implementation levels: STOWS, OntoTest, and the ontologies presented in [Arnicans et al. 2013], [Bai et al. 2008] and [Sapna and Mohanty 2011]. It is important to clarify the approach

followed in [Arnicans et al. 2013]. In this study, first the ontology is semi-automatically generated in OWL. The obtained ontology is then transformed to UML class diagram using a tool called OWLGrEd in order to be evaluated by experts.

Concerning the languages used for representing the ontologies, all the studies that present the ontologies in the implementation level used OWL. In the conceptual level, all the ontologies are presented as UML class diagrams. Moreover, two ontologies use first order logics to capture some axioms, namely OntoTest and the ontology presented in [Li and Zhang 2012].

Summarizing, from the 12 ontologies investigated, 2 are represented only as conceptual models presented as UML class diagrams (TaaS Ontology, and the ontology presented in [Li and Zhang 2012]), 5 are represented only as OWL implementations (the ontologies presented in [Guo et al. 2011], [Nasser et al. 2009], [Ryu et al. 2011], [Cai et al. 2009] and [Anandaraj et al. 2011]), and 5 are represented both in the conceptual level (as UML class diagrams) and in the implementation level (as OWL artifacts) (STOWS, OntoTest, and the ontologies presented in [Arnicans et al. 2013], [Bai et al. 2008], and [Sapna and Mohanty 2011]).

Finally, although foundational ontologies have been recognized as an important instrument for improving the quality of conceptual models in general, and more specifically of domain ontologies [Guizzardi 2007], none of the ontologies analyzed in our SLR reuses foundational ontologies.

4. Discussion

In this section, we discuss some relevant points that have arisen from the data syntheses done in the SLR and discuss limitations of them.

Currently, software testing is considered a complex process comprising activities, techniques, artifacts, and different types of resources (hardware, software and human resources). Thus, building a complete testing ontology is not a trivial task (if even possible). Although there are a relatively large number of ontologies on software testing published in the literature (at least 12 ontologies), we notice that there are still problems related to the establishment of an explicit common conceptualization regarding this domain. For being applied to KM, a software testing ontology must take some characteristics of good quality ontologies into account.

In an experiment trying mainly to identify good practices in ontology design, D'Aquin and Gangemi [D'Aquin and Gangemi 2011] have identified some characteristics that are presented in what they call "beautiful ontologies". These characteristics were grouped in three dimensions: (i) formal structure, (ii) conceptual coverage and task, and (iii) pragmatic or social sustainability. In order to evaluate the testing ontologies selected by means of the 2nd SLR, we focus on the first dimension, and in part of the second one, namely conceptual coverage. The characteristics included in these dimensions are [D'Aquin and Gangemi 2011]:

> **Structure:** the ontology reuses foundational ontologies; the ontology is designed in a principled way; it is formally rigorous; it also implements non-taxonomic relations; the ontology strictly follows an evaluation method; it is modular, or embedded in a modular framework.

Conceptual coverage: the ontology provides important reusable distinctions; it has a good domain coverage; it implements an international standard; the ontology provides an organization to unstructured or poorly structured domains.

Unfortunately, some of these characteristics are difficult to evaluate, since there isn't much information about them in the papers presenting the corresponding ontologies. Thus, in our analysis, we focused on the most easily discernible features, namely: having a good domain coverage; implementing an international standard; being formally rigorous; implementing also non-taxonomic relations; following an evaluation method; and reusing foundational ontologies.

Regarding the first characteristic (having a good domain coverage), we notice that most ontologies have very limited coverage (see Section 3.3). Those that have higher coverage are: STOWS, OntoTest, and TaaS. Some take international standards into account, namely: OntoTest, and the ontologies presented in [Arnicans et al. 2013], [Bai et al. 2008], [Sapna and Mohanty 2011], and [Cai et al. 2009]. Others, on the other hand, do not consider international standards (or at least do not mention them). This is the case of STOWS and TaaS Ontology.

The next two characteristics (being formally rigorous and also implementing non-taxonomic relations) are very important for a reference ontology. As discussed previously, a reference ontology must be a heavyweight ontology, and thus it must comprise conceptual models that include concepts, and relations (of several natures), and also axioms describing constraints and allowing to derive information from the domain models. Taking this perspective into account, we can notice that most of the existing ontologies present problems.

There are five ontologies ([Guo et al. 2011], [Nasser et al. 2009], [Ryu et al. 2011], [Sapna and Mohanty 2011] and [Anandaraj at al. 2011]) that are just OWL artifacts (i.e., operational ontologies), and thus are not enough for the purposes of applying ontologies for KM.

The ontologies presented in [Arnicans et al. 2013] and [Cai et al. 2009] are, in fact, taxonomies, and thus, in our view, they do not qualify as ontologies (or at most, they are lightweight ontologies). STOWS is mainly a set of taxonomies of basic concepts, including some properties and few relations. There are taxonomies of Tester, Context, Testing Activities, Testing Methods, and Testing Artifacts, but there are important relations missing. For instance, which are the artifacts produced and required by a testing activity? Without relations between the concepts, questions such as this one cannot be answered. Moreover, there are two "compound concepts" in STOWS that are defined on the bases of the basic concepts: capability and task. *Capability*, for instance, is modeled as a composite entity, which parts are *Activity*, *Method*, an optionally *Environment*, *Context*, and *Data* (a subtype of *Artifact*). This model is questionable, since it puts together objects and events as part of *Capability*. Objects (or endurants) exist in time; while events (or perdurants) happen in time [Guizzardi 2008]. So what is a Capability? An object or an event? This shows that this ontology presents problems.

TaaS Ontology presents very simple models. UML class diagrams presented in [Yu et al. 2008] and [Yu et al. 2009] do not specify multiplicities of relationships. Moreover, like STOWS, most of the relationships are modeled as aggregations (whole-

part relations in UML). This approach is very questionable from an ontological point of view. For instance, there is a core concept called *Test Task*, which is modeled as composed of *TestActivity*, *TestType*, *TargetUnderTest*, *TestEnvironment*, and *TestSchedule*. Analogously to the analysis on STOWS, the composite object *Test Task* aggregates endurants and perdurants.

Even the most complete ontology among the ones we achieved through the SLR, OntoTest, also presents problems. First, there are sub-ontologies that were not published yet, namely the Testing Process, Testing Phase, Testing Artifact, and Testing Procedure sub-ontologies. Second, OntoTest does not properly link the concepts in the sub-ontologies. For instance, albeit in the Main Software Testing Ontology there is a relationship between Testing Step and Test Resource, there aren't relationships between their subtypes. This is an important part of the software testing conceptualization that needs to be made explicit.

Regarding ontology evaluation, none of the works we investigated in the SLR discusses how the ontologies they propose were evaluated, except the one done by Arnicans et al. (2013), which says that a software testing expert has analyzed the ontology fragment related to testing techniques.

Finally, concerning the reuse of foundational ontologies, none of the ontologies analyzed in our SLR have used one. In our view, this is a problem, because important distinctions made in Formal Ontologies may be disregarded as clearly noticed in the brief analysis we did (as in the aforementioned cases of STOWS and TaaS Ontology). The lack of truly ontological foundations puts in check the truthfulness of those ontologies.

Thus, we concluded that the software testing community has still a lot work to do, in order to advance towards a reference software testing ontology. Once developed a good quality reference testing ontology, an operational version of it should be designed and implemented. With these two artifacts in hand, we can effectively take a step forward in ontology-based KM applied to the software testing domain.

Limitations of the SRL

The SLR presented in this paper has some limitations. Due to the fact that the study selection and data extraction steps were performed by just one of the authors, some subjectivity may have been inserted. To reduce this subjectivity, the other two authors performed these same steps in a random sample (including about 35% of the studies). The results of each reviewer were then compared in order to detect possible bias. Moreover, terminological problems in the search strings may have led to missing some primary studies. In order to minimize these problems, we performed previous simulations in the selected databases. We decide not to search any specific conference proceedings, journals, or the grey literature (technical reports and works in progress). Thus, we have just worked with studies indexed by the selected electronic databases. The exclusion of these other sources makes the review more repeatable, but possibly some valuable studies may have been left out of our analysis.

5. Conclusions

In this paper, we presented a Systematic Literature Reviews (SLR) in order to investigate ontologies in the software testing domain, including questions related to their coverage of the software testing domain, and how they were developed. We identified 12 ontologies addressing the software testing domain. For analyzing these ontologies, we considered some of the characteristics pointed by D'Aquin and Gangemi (2011) as characteristics that are presented in "beautiful ontologies". In our analysis, we considered the following characteristics: having a good domain coverage; implementing an international standard; being formally rigorous; implementing also non-taxonomic relations; following an evaluation method; and reusing foundational ontologies.

As the main findings obtained from this SLR, we highlight the following conclusions: most ontologies have limited coverage; the studies do not discuss how the ontologies were evaluated; none of the analyzed testing ontologies is truly a reference ontology; and none of them is grounded in a foundational ontology. In sum, we conclude that the software testing community should invest more efforts to get a well-established reference software testing ontology.

As a future work, we intend to develop a KM system for managing testing-related knowledge items. This KM system will be built based on a Reference Ontology on Software Testing (ROoST) that we are now developing [Souza 2013].

Acknowledgments - The first author acknowledges FAPESP (Process: 2010/20557-1) for the financial grant. The second author acknowledges FAPES/CNPq (PRONEX Grant 52272362/11) for the financial grant.

References

Abecker, A., van Elst, L. (2009). Ontologies for Knowledge Management, In: Handbook of Ontologies, Staab, S., Studer, R. (Eds.), Springer, 2nd edition.

Andrade, J., Ares, J., Martínez, M., Pazos, J., Rodríguez, S., Romera, J., Suárez, S. (2013). An architectural model for software testing lesson learned systems, Information and Software Technology. vol. 55, 18-34.

Anandaraj, A., Kalaivani, P., Rameshkumar, V. (2011). Development of Ontology-Based Intelligent System for Software Testing. In. International Journal of Communication, Computation and Innovation, v. 2.

Arnicans, G., Romans, D., Straujums, U. (2013). Semi-automatic Generation of a Software Testing Lightweight Ontology from a Glossary Based on the ONTO6 Methodology," In Frontiers in Artificial Intelligence and Applications, 263-276.

Bai, X., Lee, S., Tsai, W., Chen, Y. (2008). Ontology-Based Test Modeling and Partition Testing of Web Services. International Conf. on Web Services, 465-472.

Barbosa, E. F., Nakagawa, E. Y., Maldonado, J. C. (2006). Towards the establishment of an ontology of software testing. In. International Conference on Software Engineering and Knowledge Engineering (SEKE), v.1, 522-525, San Francisco, CA.

Barbosa, E. F., Nakagawa, E. Y., Riekstin, A. C., Maldonado, J. C. (2008). Ontology-based Development of Testing Related Tools. In International Conference on Software Engineering & Knowledge Engineering (SEKE), San Francisco, CA.

Cai, L., Tong, W., Liu, Z., Zhang, J. (2009). Test Case Reuse Based on Ontology. Pacific Rim International Symposium on Dependable Computing, 103-108.

Davies, J., Fensel, D., Van Harlemen, F. (2003). Towards the Semantic Web: Ontology-driven Knowledge Management, John Wiley & Sons.

Dyba T., Dingsoyr, T., Hanssen, G. (2007). Applying systematic reviews to diverse study types: An experience report. First International Symposium on Empirical Software Engineering and Measurement, Madrid, 225-234.

D'Aquin, M. and Gangemi, A. (2011). Is there beauty in ontologies? Applied Ontology. vol. 6, n.3, 165-175.

Falbo, R. A., Menezes, C. S. and Rocha, A. R. (1998). A systematic approach for building ontologies. In VI Ibero-American Conference on AI (IBERAMIA98), Lisboa, Portugal.

Guizzardi, G. (2007). On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In: Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV, 18–39, Amsterdã.

Guizzardi, G., Falbo, R.A., Guizzardi R.S.S. (2008). Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology. In XI Iberoamerican Workshop on Requirements Engineering and Software Environments, 244-251.

Guo, S., Zhang, J., Tong, W., Liu, Z. (2011). An Application of Ontology to Test Case Reuse. In. International Conference on Mechatronic Science, Electric Engineering and Computer, 19-22, Jilin, China.

Hong, Z. (2006). A Framework for Service-Oriented Testing of Web Services. Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International, Chicago, IL, 145 - 150.

Huo, Q., Zhu, H., Greenwood, S. (2003). A Multi-Agent Software Environment for Testing Web-based Applications. In 27th International Computer Software and Applications Conference (COMPSAC2003), Dallas, TX, USA, 210-215.

IEEE Std 1012 (2004). IEEE Standard for Software Verification and Validation. New York, NY, USA.

IEEE Computer Society, SWEBOK. (2004). A Guide to the Software Engineering Body of Knowledge. <http://www.computer.org/portal/web/swebok>

Juristo, N., Ferndandez, M., Gomez-Perez, A. (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In Proceedings of the AAAI97 Spring Symposium. Technical Report SS-97-06, 15(2).

Kitchenham, and Charters, B. S. (2007). EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics Keele University and Departament of Computer Science University of Durham, UK, v. 2.3.

Li, X. and Zhang, W. (2012). Ontology-based Testing Platform for Reusing. In. Internet Computing for Science and Engineering (ICICSE 2012), 86 – 89.

Mathur, A. P. (2012). Foundations of Software Testing. 5th ed. Delhi, India: Dorling Kindersley (India), Pearson Education in South Asia.

Nakagawa, E.Y., Barbosa, E.F., Maldonado, J.C. (2009). Exploring ontologies to support the establishment of reference architectures: An example on software testing. Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on, Cambridge, 249 - 252.

Nasser, V. H., Du, W., MacIsaac, D. (2009). Knowledge-based software test generation. In. International Conference on Software Engineering and Knowledge Engineering, SEKE'2009, 312 - 317.

Noy, N.F., McGuinness, D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI.

O´Leary, D.E. (1998). Enterprise Knowledge Management, Computer, Univ. of Southern California, Los Angeles, CA, v. 31, Issue 3, 54 – 6.

Ryu, H., Ryu, D., Baik, J. (2011). A Strategic Test Process Improvement Approach Using an Ontological Description for MND-TMM. In. International Conference on Computer and Information Science, 561-566.

Sapna, P. G. and Mohanty, H. (2011). An Ontology Based Approach for Test Scenario Management. ICISTM'2011, v. 141, 91–100.

Souza, E. F., Falbo, R. A., Vijaykumar, N. L. (2013). Using Ontology Patterns for Building a Reference Sofware Testing Ontology. In: The 8th International Workshop on Vocabularies, Ontologies and Rules for the Enterprise and Beyond (VORTE2013). The 17th IEEE International EDOC Conference (EDOC2013), Vancouver, BC.

Uschold, M. and King, M. (1995). Towards a Methodology for Building Ontologies.. Presented at the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI95, AIAI-TR-183, University of Edinburgh, Edinburgh.

Yufeng, Z. and Hong, Z. (2008). Ontology for Service Oriented Testing of Web Services. Symposium on Service-Oriented System Engineering, Jhongli, 129 - 134.

Yu, L., Su, S., Zhao, J. (2008). Performing Unit Testing Based on Testing as a Service (TaaS) Approach. In. International Conference on Service Science, 127-131.

Yu, L., Zhang, L., Xiang, H., Su, Y., Zhao, W., Zhu, J. (2009). A Framework of Testing as a Service. Management and Service Science, International Conf. on, 1- 4.

Zhu, H. and Huo, Q. (2005). Developing A Software Testing Ontology in UML for a Software Growth Environment of Web-Based Applications. Software Evolution with UML and XML, 263-295, IDEA Group.

Zhu, H. and Zhang, Y. (2012). Collaborative Testing of Web Services. In. IEEE Transactions on Service Computing, 116 - 130, v. 5.

Semantic Search Architecture for Retrieving Information in Biodiversity Repositories

Flor K. Amanqui[1], Kleberson J. Serique[1], Franco Lamping[1], Andréa C. F. Albuquerque2[2], José L. C. Dos Santos[2], Dilvan A. Moreira[1]

[1]University of São Paulo (USP) – CEP: 13566-590 – São Carlos – SP – Brazil

[2]National Institute for Amazonian Research (INPA)
CEP.: 69060-001 – Manaus – AM – Brazil

{flork,serique,dilvan}@icmc.usp.br, lamping@grad.icmc.usp.br

andreaalb.1993@gmail.com, lcampos@inpa.gov.br

Abstract. *The amount of biological data available electronically is increasing at a rapid rate; for instance, over 16.500 specimens are available today in the National Institute for Amazonian Research (INPA) collections. However, this data is not semantically categorized and stored and thus is difficult to search. To tackle this problem, we present a semantic search architecture, implemented using state of the art semantic web tools, and test it on a set of representative data about biodiversity from INPA. This paper describes how the mechanism of mapping is designed so that the semantic search can find information, based on ontologies. We show a series of SPARQL queries and explain how the mapping mechanism works. Our experiments, using a prototype of the proposed architecture, showed that the prototype had better precision and recall then traditional keyword based search engines.*

Keywords: Biodiversity, Ontology, Data Integration, Semantic Search

1. Introduction

Biological diversity, or biodiversity, is the term given to the variety of life on Earth. Biodiversity is the combination of life forms and their interactions with one another, and with the physical environment that has made Earth habitable.

The biodiversity information that can be obtained via Internet continues to grow significantly. Every day, new collections, databases, and applications are being added. This information is stored in a variety of formats (spreadsheets, html, xml, pdf and catalogues, amongst others). This proliferation of information from different sources means that the search for information could be met by a variety of available resources, which store data about the same domains but have different characteristics. For that reason, much of this information is never found. The need for integration and analysis of biodiversity information becomes evident.

In this context, finding relevant and recent information is a hard task that is not particularly well supported by current biodiversity software tools. Keyword-based search have serious problems associated with its use: low or no recall; high recall, low precision; initial keywords in search often do not get the wanted results.

The semantic web (an extension of the current Web) tries to represent information in such a way that it can be used by machines, not just for display purposes, but also for automation, integration and reuse across applications [Boley et al. 2001].

There are a number of important technologies related to the Semantic Web: ontologies, languages for the Semantic Web, semantic search, semantic markup of pages and services (that the Semantic Web is supposed to provide). Ontologies, one of the most important ones, are implemented in the RDF(S) (Resource Description Framework/Schema) and OWL (Web Ontology Languages) languages, two W3C recommended data representation models.

In this article, we propose a semantic search architecture that supports mapping between biodiversity data, from INPA's (National Institute for Amazonian Research) collections, stored in relational databases and the ontologies describing it.

The rest of this article is organized as follows: Section 2 describes related works. Section 3 describes our biodiversity ontology. Section 4 presents our semantic search architecture. Section 5 presents a synopsis of our experiments and Section 6 concludes by summarizing our results and describing future works.

2. Related Works

Researchers have proposed various techniques and approaches designed to perform semantic search. We studied a number of them that could be used in the area of biodiversity.

In [Xiong et al. 2009], a method of search based on a smart query agent is proposed (Geoonto). It retrieves information from data catalogs/databases using ontologies. This method associates semantic information in the search process, and generates a refined query string.

In [Latiri et al. 2012], an automatic method of query expansion is proposed in which user requests are expressed in natural language.

In [Mittal et al. 2010], a method hybrid of personalized web information is proposed in which ontology for retrieval of user context is used and a user profile is being maintained.

In [Li and Yang 2008], a method to construct a semantic search engine is proposed. It provides a uniform platform to search, view and operate spatial on information.

In [Santos et al. 2011], an architecture to support semantic search in a metadata repository is proposed. This work discover similar concepts even when different terms are used in their designation or description, since a domain ontology is used to annotate information sources and to expand the user query with terms from the universe of discourse.

A number of techniques have been developed for using ontologies to retrieve relevant documents in response to a query. However, none of them focused on the problem of storage and retrieval of RDF triples. Most of these techniques require complex analysis, involving natural language processing, to discover the context and semantics of query terms. Also, an additional limitation, in many of the existing approaches, is the lack of a quality evaluation of results.

We have developed a semantic search application that uses key semantic web concepts for information retrieval and also technologies such as mapping, triple store and SPARQL queries.

3. The Biodiversity ontology

OntoBio is a biodiversity ontology developed by INPA and UFAM (Federal University of Amazonas) and extended by USP (University of São Paulo). Its main objective is to provide a clear and precise conceptualization of the aspects considered in biodiversity data collection, regardless of a specific application.

The original version of OntoBio is presented in details at [Albuquerque 2011]. One of the advantages of having data annotated using OntoBio concepts is that it can be reused as Linked Data. Linked Data describes a method of publishing structured data so that it can be interlinked and become more useful [Kauppinen and de Espindola 2011].

To better archive that, data annotated using OntoBio has to be easily interlinked with other biodiversity data, already available on the web (as part of the wider Linked Data community), through the use of as many shared concepts as possible. With that in mind, we rewrote the first version of OntoBio to reuse, whenever possible, terms from other public available ontologies to allow better "linkability" with data already annotated using them.

We added terms from the following public ontologies:

- The Phenotypic Quality Ontology[PATO 2010], which is an ontology of phenotypic qualities, intended for use in a number of applications, primarily defining composite phenotypes and phenotype annotation;
- Basic Geo Vocabulary [WGS84 2003] is a basic RDF vocabulary that provides the semantic web community with a namespace for representing lat(itude), long(itude) and other information about spatially-located things, using WGS84 as a reference datum, and;
- The Geoname Ontology[GeoNames 2011] makes it possible to add geospatial semantic information to the Word Wide Web. All over 8.3 million geonames toponyms now have a unique URL with a corresponding RDF web service.

The OntoBio ontology is presented in the Figure 1. The Protégé 4 ontology editor was used to write the OntoBio ontology in OWL 2 DL. The new version of OntoBio is available through the NCBO's Bioportal http://bioportal.bioontology.org/ontologies/50517. There, users can download, browse and suggest terms for the ontology.

4. An Architecture for Semantic Search

We have proposed the architecture of a semantic search that follows the mechanism of mapping between OntoBio domain ontology, and Database from INPA the collections of insects, fishes, and mammals.

The system overall architecture is shown in Figure 2. It consists of four basic modules: User Interface Layer, Query Reformulation, Mapping Component and Data Access Layer.

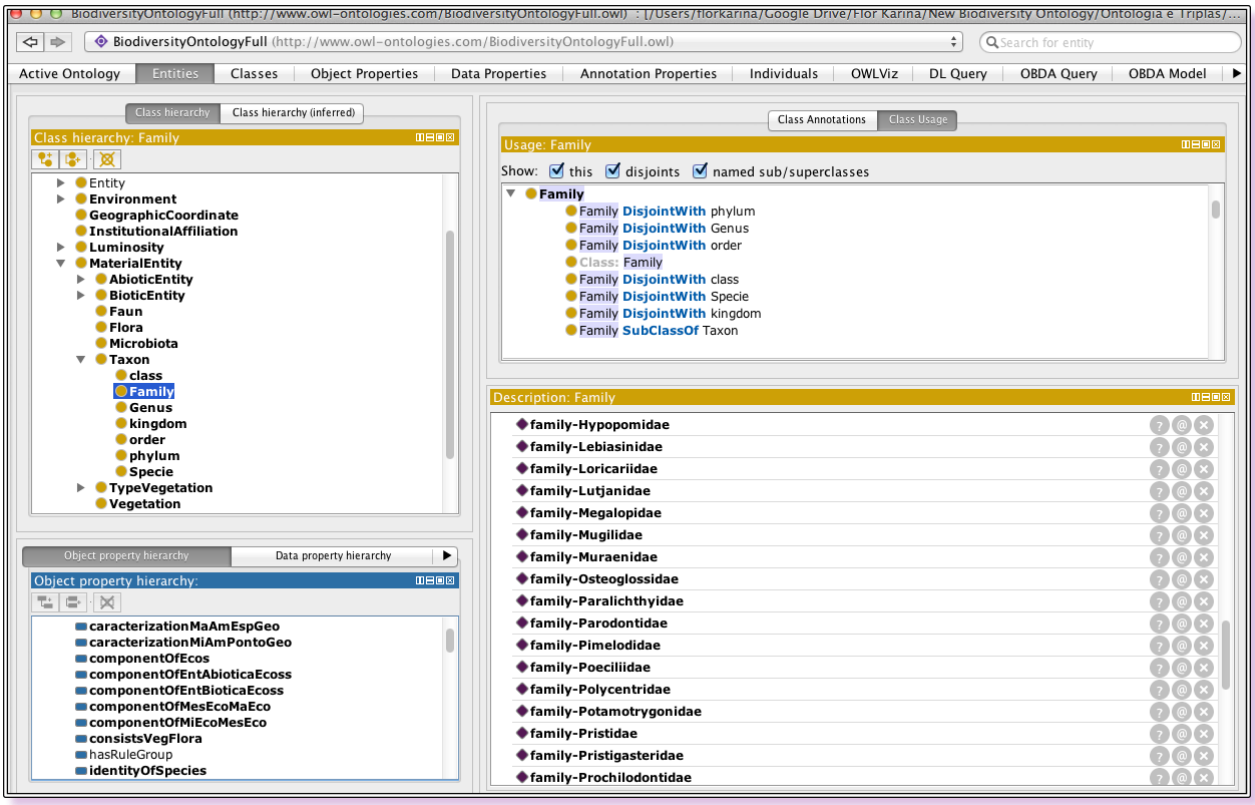


Figure 1. New version of biodiversity ontology

1. **User Interface Layer** is responsible for the interaction between users and system. The search process begins with an initial keyword list, entered by the users, that represents his/her search intentions.
2. **The Query Reformulation component** receives the input of search terms from the user, selects and expands keyword lists by adding semantically related terms, using techniques of expansion and semantic similarity. It uses the SPARQL Writer component to take keyword lists and generate SPARQL queries from them. It uses an algorithm that will be described on the following sections.
3. **The Mapping Component** loads the domain ontologies, taxonomic information and the collection database and transforms them in a set of Resource Description Framework (RDF) triples. We used Ontop, a platform to query databases as Virtual RDF Graphs using SPARQL, to do the mapping between the relational databases records and the OWL ontologies.
   Ontop is a platform to query databases as Virtual RDF Graphs using SPARQL. It does the mapping between the relational databases records and the OWL ontologies. Ontop has two tools: OntopPro, which is a Protege 4 plugin that implements a graphic mapping editor; and Quest, which is a SPARQL query engine/reasoner that supports RDFS and OWL 2 QL entailment regimes and SPARQL-to-SQL query rewriting (Mariano R and Calvanese, 2012). The mapping process is divided in three steps:
   (a) **Creation of Mapping Axioms:** OntopPro mappings are done using mapping axioms. A mapping axiom is defined by an SQL query and an ABox

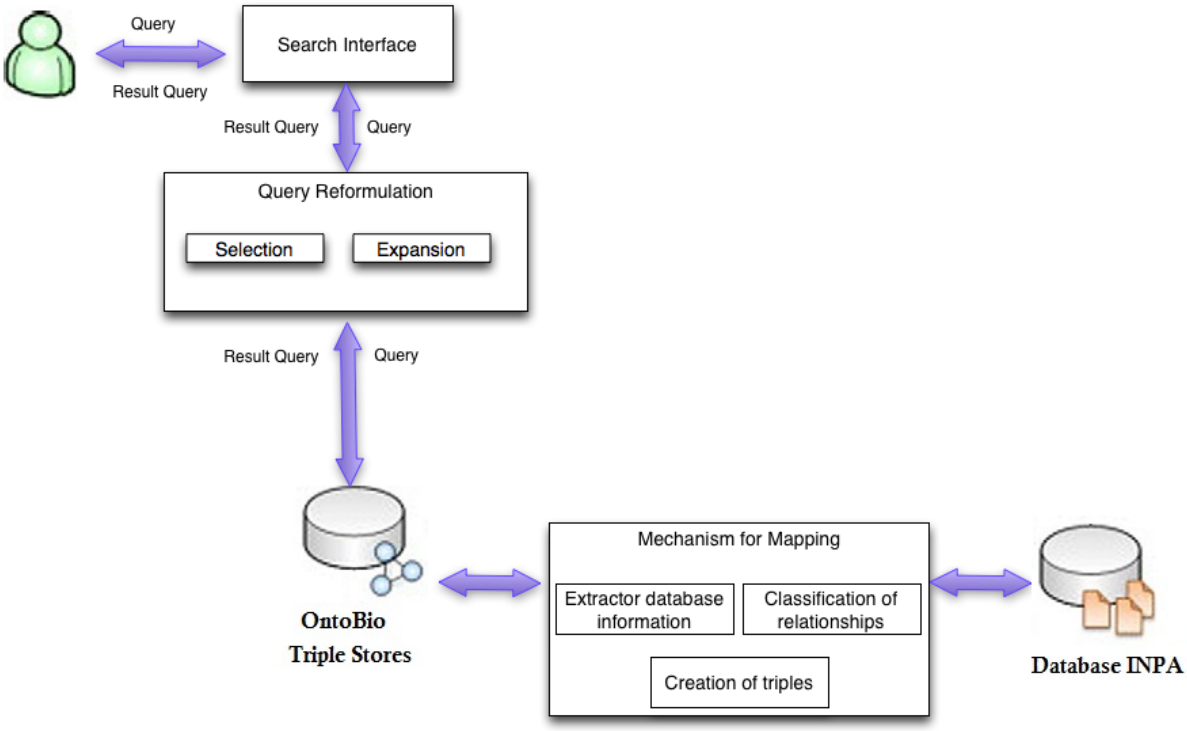


Figure 2. Architecture for Semantic Search

assertion template (Figure 3). An Abox assertion template is a set of RDF/OWL triples, written in a turtle-like syntax, in which the subject and object of the triples allow for variables that reference columns of the SQL query result [Mariano R and Calvanese 2012].
In other words, a mapping axiom defines how the values in each row of the results (of an SQL query) can be used to generate a set of ABox assertions. The mapping axioms were created using information from the OntoBio ontology and INPA experts. Each mapping must contain one or more mapping axioms. Figure 3 shows a valid mapping.

(b) **Generation of RDF Triples:** Mapping axioms generate RDF triples. This generation is done using the Quest tool from Ontop. The Quest reasoner uses query-rewriting techniques to generate triples. The triples are created by replacing the placeholders in the target with the values from the SQL row.

(c) **RDF Triples Loader:** Using OntopPro, it is possible to export the RDF triples generated by the Quest tool to a file. That file is then loaded into the Virtuoso triple store, which is now ready to answer queries using them. The Mapping Component can repeat the process described here, whenever INPA releases updates to its collection records.

4. **Data Access layer** that is the architecture layer that provides access to the RDF triples stored in the Virtuoso Triple Store, using SPARQL, both for the layer above it and for other machines on the network. Triple Store is the common name given to a database management system for RDF Data.

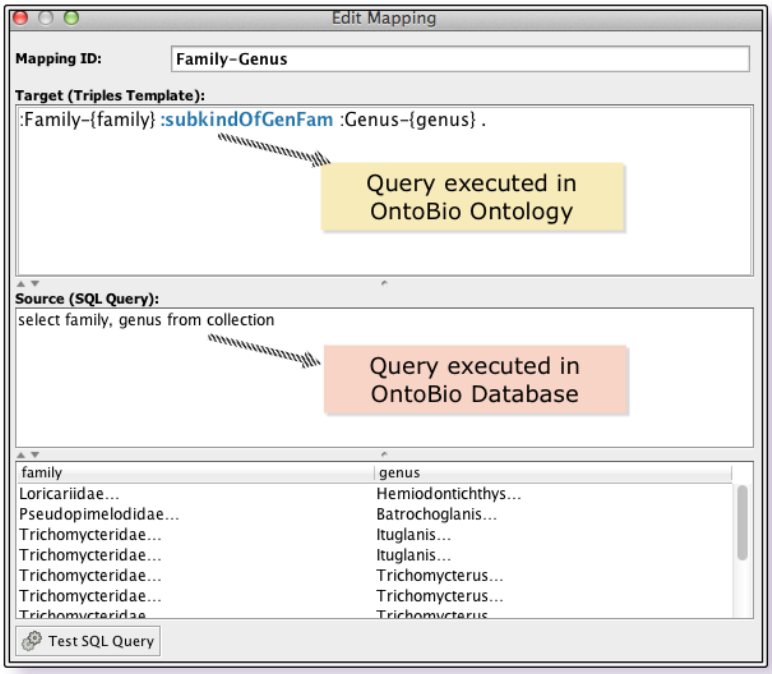


Figure 3. Mapping axiom

4.1. Semantic Search Algorithm

The basic idea of our algorithm is to compare input keywords with OntoBio resources (subject, predicate and object) in the Virtuoso triple store. The Virtuoso platform was chosen because it can store the triples generated from INPA data and work with multiple graphs at the same time.

```
I/P – String nameClass, name of the class selected by user
O/P – String result, result of the queries
BEGIN
Step 1: Establish connection with Virtuoso Triple Store
and ontology OntoBio
Step 2: extracts OntoBio information and find hierarchy
Attribute initialNameClass as nameClass
    While (nameClass has SuperClass)
    BEGIN
    Step 3: Submit SPARQL query with query user as object and find
    the SuperClass (subject) comparing similarity with predicates
    Step 4: Attach result
    END
Step 5: Submit SPARQL query with initialNameClass as subject and
find the geographical location (subject, predicate and object).
Step 6: Attach result
Step 7: Return result
End
```

We implemented this algorithm in a prototype using: Java, Eclipse Indigo (as IDE), Google Web Toolkit 2.5.1 to create a web client, Jena RDF framework to process (simplified) SPARQL queries and Virtuoso Server as triple store.

Figure 4 shows graphic interface to support user queries. We implemented a SPARQL Endpoint for INPA http://143.107.231.220:8890/sparql and implemented a set of queries described on experiments section.

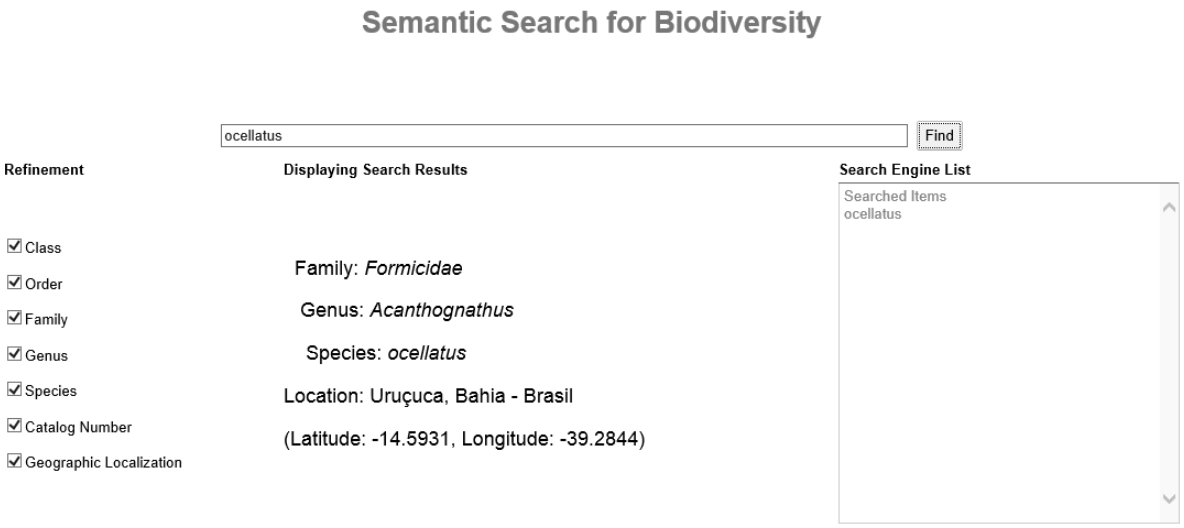


Figure 4. Web application for searching biodiversity information

5. Experiments

In order to validate our proposed architecture, researchers from our group and biodiversity scientists were interviewed to categorize important information from the INPA data.

We defined use cases (Table 1) with scenarios to identify the various user tasks and built SPARQL queries related with these use cases.

For each of the previous use cases, biodiversity experts identified the information set each user needed for each task and examples of queries that should have returned this information. After we tested each query, the same experts judged which results were relevant and non relevant (relevance non relevance judgment).

This process of information feedback is commonly referred to, in the literature, as relevance feedback [Salton 1971] when experts explicitly provide information on relevant documents to a query [Baeza-Yates and Ribeiro-Neto 1999]. In its original formulation, expert users inspect the query results and indicate those that are really relevant to the search. Table [tab:InfoNeeds] shows examples of users tasks and possible query strings to get the relevant biodiversity information.

Scientists can identify species using the taxonomic classification system no matter what their language. The taxonomic classification system is composed by a hierarchy (series of ranks) that shows the kinship of organisms and also, whenever possible, ancestor-descendant relationships.

Table 1. Biodiversity Use Cases

Use Cases	Goals	Queries
Use Case 01	Identification of a species.	Query1: fish ocellatus Query2: fish brasiliensis Query3: fish Corydoras splendens
Use Case 02	Determine information of a collect.	Query4: fish Hemigrammus gracilis Query5: Potamorrhaphis guianensis Query6: Hemigrammus guyanensis Query7: Iguanodectes spilurus
Use Case 03	Determine the best areas for aquaculture considering different types of species and geographical location of a collect.	Query8: Gnathocharax steindachneri

The basic ranks of the taxonomic classification system are kingdom, phylum, class, order, family, genus and species. The following SPARQL query (Listing 1) shows taxonomic system of classification for the *kingdom Animalia*.

Listing 1. SPARQL query returning the taxonomy of a specie

```
PREFIX oo: <http://www.owl-ontologies.com/
BiodiversityOntologyFull.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?phylum ?class ?order ?family ?genus ?species
where { oo:kingdom-Animalia oo:subkinofPhyKing ?phylum.
?phylum oo:subkinofClassPhy ?class .
?class oo:subkinofOrdClass ?order .
?order oo:subkinofFamOrd ?family .
?family oo:subkindOfGenFam ?genus .
?genus oo:subkindOfEspGen ?species .
}
```

The following SPARQL query (Listing 2) shows important information from a collect such as Collect, Research Institution, Method, Determinate Name.

Listing 2. SPARQL query returning information of a collect

```
PREFIX : <http://www.owl-ontologies.com/
BiodiversityOntologyFull.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?collect ?ResearchInstitution ?MethodCollect
?NameDeterminateCollect where {
?collect :mediationInstituicaoVinculo ?ResearchInstitution.
?collect :isClassifiedAsColetaTipoColeta ?MethodCollect .
?collect :mediationColetaRespColeta ?NameDeterminateCollect.}
```

The following SPARQL query (Listing 3) shows the geographical location of a specimen collect and other data, such as collect local, geographic space, latitude and longitude.

Listing 3. SPARQL query returning geographical location of a collect

```
PREFIX : <http://www.owl-ontologies.com/
BiodiversityOntologyFull.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?CollectLocal ?GeographicSpace ?latitude ?longitude
where {
?CollectLocal :localizationEspaGeoCoordGeo ?GeographicSpace.
?GeographicSpace :latitude ?latitude .
?GeographicSpace :longitude ?longitude .
}
```

To evaluate our semantic search architecture, we measured precision and recall to assess the performance of each approach dependent on input variable such as the user query. The recall value measures whether a tool retrieves all possible items related to the search terms contained in the data store, while precision measures to what extent only the relevant items were actually returned.

We compared the result in two search systems, our semantic search and keyword based search from SpeciesLink with data from INPA. We used a total of 16 queries (8 for each system).

To compare the results of only two systems, we will employ the Students T-tests, since they are designed for testing two data sets [B. Rasch and Naumann 2004]. When checking two data sets, each characterized by its average, standard deviation, and number of data points, it is possible to apply the T-test to identify, whether the means are in fact distinct or not. A probability value (p-values) below 0.05 indicates a statistically significant difference, whereas a p-value equal or exceeding 0.05 indicates no significant evidence, that there exists no significant difference between the performance values of two or more tools [Sachs 2003].

In our experiments, Semantic Search resulted in is significant difference in recall (p=0.0201 by t-test) and precision (p= 0.0006 by t-test) when compared to Keyword based search. One reason might be that keyword based search is not enough to capture the underlying semantics of user information needs, since it is content-oriented. This evaluation is shown in Table 2.

There is a significant difference in the mean of precision in Semantic Search minus the mean precision in Keyword Search equals 0.50416. The confidence interval of this difference from 0.302420283 to 0.705913042 is 95%. The mean of recall in Semantic Search minus the mean precision in Keyword Search is equals 0.20624745663. The confidence interval of this difference from 0.04330054489 to 0.36919436836 is 95%.

Table 2. Students T-tests

Group	Semantic Search (Recall)	Keyword Based Search (Recall)	Semantic Search (Precision)	Keyword Based Search (Precision)
Mean	0.587638057	0.3813906	0.975	0.470833338
Queries	8	8	8	8

6. Conclusions and Future Work

The architecture presented in this work provides a new document retrieval process by exploiting query terms to support scientists in the process of discovery and integrating biodiversity data and domain knowledge. This architecture can be classified, according to the categorization schema proposed by [Mangold 2007], as a Stand-Alone Search Engine. The search process uses resources labels from classes, properties, mappings and instances from domain ontologies represented in the OWL language.

We defined a mapping mechanism between relational database data and OntoBio ontology terms resulting in the generation of RDF triples (subject, object and predicate) saved in a triple store (Virtuoso). The triple stores make it much easier to add new predicates and write complicated queries or perform inferencing and rule processing.

A comparative analysis showed a significant increase in recall and precision in the semantic search. The possibility of creating queries that seek information based on relationships between data offers many alternatives to semantic search systems, since the results of these queries are not based only on specific information. Users can thus receive data that, in traditional systems, would not be considered by the query, but by analyzing their relations with other information, semantic search queries can consider them relevant.

As future work, we intend to extend our current implementation with more advanced structured searches in partnership with researches from INPA.

7. Acknowledgment

The authors would like to thank INPA for supporting this work. Thanks are also due to researchers of INPA's biological collections. This research was financed by the Brazilian funding agency CNPq.

References

Albuquerque, A. (2011). *Desenvolvimento de uma Ontologia de DomÃnio para Modelagem de Biodiversidade*. DissertaÃ§Ã£o de Mestrado. Universidade Federal do Amazonas.

B. Rasch, M. Friese, W. H. and Naumann, E. (2004). *Quantitative Methoden Band*. Springer, ISBN 978-3-540-33307-4.

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Boley, H., Tabet, S., and Wagner, G. (2001). Design rationale of ruleml: A markup language for semantic web rules. pages 381–401.

GeoNames (2011). Geonames ontology. `http://www.geonames.org/ontology/documentation.html`. Accessed: 2013-07-30.

Kauppinen, T. and de Espindola, G. M. (2011). Linked Open Science—communicating, sharing and evaluating data, methods and results for executable papers. *Proceedings of the International Conference on Computational Science (ICCS 2011), Procedia Computer Science*, 4(0):726–731.

Latiri, C. C., Haddad, H., and Hamrouni, T. (2012). Towards an effective automatic query expansion process using an association rule mining approach. *J. Intell. Inf. Syst.*, 39(1):209–247.

Li, W. and Yang, C. (2008). A semantic search engine for spatial web portals. volume 2, pages II–1278 –II–1281.

Mangold, C. (2007). A survey and classification of semantic search approaches. *Int. J. Metadata Semant. Ontologies*, 2(1):23–34.

Mariano R, M. and Calvanese, D. (2012). *Quest, an OWL 2 QL Reasoner for Ontology-Based Data Access*. KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Bolzano, Italy.

Mittal, N., Nayak, R., Govil, M. C., and Jain, K. (2010). A hybrid approach of personalized web information retrieval. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 308–313.

PATO (2010). The phenotypic quality ontology. `http://bioportal.bioontology.org/ontologies/1069`. Accessed: 2013-07-30.

Sachs, L. (2003). *Angewandte Statistik: Anwendung statistischer Methoden*. Springer, November. ISBN 3540405550.

Salton, G., editor (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, New Jersey.

Santos, V. D., Baiao, F. A., and Tanaka, A. (2011). An architecture to support information sources discovery through semantic search. In *Information Reuse and Integration*.

WGS84 (2003). W3C Semantic Web Interest Group: Basic Geo (WGS84 lat/long) Vocabulary.

Xiong, J., Huang, W., and Jin, C. (2009). An ontology-based semantic search approach for geosciences. volume 3, pages 87 –90.

Merging Ontologies via Kernel Contraction

Raphael Cóbe, Fillipe Resina, Renata Wassermann

[1]Institute of Mathematics and Statistics – University of São Paulo (USP)
São Paulo – SP – Brazil

{rmcobe,fmresina,renata}@ime.usp.br

Abstract. *Ontologies have been largely used to represent terminological knowledge, specially with the advent of the Semantic Web. As knowledge is not static, it is crucial to learn how to deal with ontology dynamics, which includes Ontology Merging and Debugging. Basically, we want to deal with the inconsistencies and incoherences that may occur when a knowledge base receives a new information or when two or more ontologies are merged. Our purpose in this work is to show some ways to extend and use the BContractor framework, originally proposed for Belief Revision, to implement operations in ontologies.*

1. Introduction

The problem of knowledge representation has been an important object of study for many years, specially in Philosophy, Artificial Intelligence and Cognitive Science. In this context, we find ontologies as an important conceptual model because they provide a formal representation of the domain they describe.

There has been a rapid increase in availability of (semantic) information on the web, i.e., ontologies. Nevertheless, there is no standard way of reusing them, creating a challenge of building new ones. This has forced users to build them from scratch instead of being able to reuse previously established ones. It has been known for years that such reuse would not come for free. The integration of multiple-source ontologies may result in conflicting information being joined together in a single ontology. This kind of problem may compromise the integrity and reliability of an ontology. Such problem is addressed from two main points of view: the theoretical and the pragmatic ones. In the theoretical field, research is more concerned about dealing with logical problems like consistency/coherence checking and solving.

Unfortunately, pragmatic approaches did not follow the same evolution pace and just a few tools have been developed to provide knowledge base integration. Most of them are not able to capture logical problems such as inconsistency and incoherence.

The Belief Revision field deals with the idea that knowledge is not set in stone, i.e., it changes along with time. So, research in the field has been investigating ways to define how an agent should accommodate new information to his/her knowledge base in a consistent way, which leads to the definition of operations to perform this task. Restrictions are given by *r*ationality postulates, which are the properties any operator should obey. Different mathematical constructions are used for the operations. *R*epresentation theorems provide the link between the rationality postulates and the constructions, showing their equivalence.

BContractor [Lundberg et al. 2012] provides a flexible framework for these operations. According to the author's definition, BContractor is "a simple, yet powerful,

interface for operations over belief bases. Simple due to an easily implementable interface. Powerful because it is extensible and easily adapted." The main contribution of this work is to show how we extended the BContractor framework in order for it to work with Description Logics and use it to solve the problem of Belief Revision that emerges from Ontology Merging.

Section 2 brings theoretical background about Belief Revision and Merging. Section 3 introduces the problem of Ontology Merging and the strategies to solve it. Section 4 presents BContractor and the adaptations we developed to extend the framework in order to make it possible to be used with Description Logics and OWL. It includes the application in Merging and the integration with the framework. Section 5 shows a small usage example in Ontology Merging.

2. Belief Revision and Merging

In this section, we briefly present the needed background in Belief Revision and Merging.

Belief Revision deals with the problem of belief dynamics. In this paper, we are interested in the application of this theory in ontology dynamics, i.e., in accommodating new information in a consistent way and also in removing some information from a knowledge base. Most of the studies in this sub-field of knowledge representation are based on the AGM paradigm, whose name derives from the initials of the authors of the seminal paper [Alchourrón et al. 1985]. This paradigm is a theory about how highly idealized rational agents should revise their beliefs when receiving new information.

The epistemic state of an agent can be represented in different ways. In the AGM paradigm the beliefs of an agent are represented by a belief set, i.e., a logically closed set of sentences. So, if K is a belief set, $K = Cn(K)$, where Cn is a supraclassical consequence operator[1]. In addition, from that paradigm we have three main operations regarding a belief set K and a sentence α: expansion (+), contraction (-) and revision(*). We use expansion when we want to simply add a new information to the set ($K + \alpha$). Contraction is used when we want to remove some information ($K - \alpha$) and revision when we want to consistently add a new information to the agent's epistemic state ($K * \alpha$).

In terms of representation, instead of belief sets we are going to represent the epistemic state of the agent by means of belief bases [Hansson 1991], which are sets not necessarily closed under logical consequence. Among the advantages of using this approach, we can cite that working with belief bases is more practical from the computational point of view, considering that belief sets are usually infinite. Moreover, in belief bases we distinguish explicit knowledge from inferred knowledge, exactly because of the absence of logical closure.

In belief base operations, we also work with expansion, contraction and revision. Expansion in bases is defined as $B + \alpha = B \cup \{\alpha\}$. In his paper [Levi 1977], Isaac Levi proposed a process for obtaining the result of a revision by means of a sequence of a contraction and an expansion. Such process became known as the *Levi identity* and works

[1]If $\mathcal{L}$ is a logic closed under the logic connectives ($\wedge, \vee, \neg, \rightarrow$), a consequence operator Cn satisfies supraclassicality if, for any $A \in 2^{\mathcal{L}}$, if α can be derived from A by classical truth-functional logic, then $\alpha \in Cn(A)$

as follows: considering * as a revision function, we formally have $B * \alpha = (B - \neg\alpha) + \alpha$. Therefore, we are going to present here only one operation, contraction. For further details about the relation between contraction and revision, see [Gärdenfors 1988].

In the AGM paradigm, contraction operations are restricted by the so-called rationality postulates. The main constructions found in the literature come equipped with representation theorems. In this paper, we are going to focus on the constructions and their implementation. For details on the postulates and representation results, please refer to [Hansson 1999]. For now, it is enough to know that we want to construct operations that, given a belief base B and a formula α, return a new belief base B' contained in B, that does not imply α and that keep as much information as possible.

In the following, we present two classical constructions for contraction, partial meet contraction [Alchourrón et al. 1985] and kernel contraction [Hansson 1999].

2.1. *Partial Meet* Contraction

A Partial Meet contraction [Alchourrón et al. 1985] of a base B by α consists in finding the maximal subsets of B that do not imply α and take the intersection of a selection of them. For this operation, we need to define the concept of a remainder set $(B \perp \alpha)$:

Definition 1 (Remainder Set) [Alchourrón et al. 1985] *Let B be a belief base and α a sentence. A set B' is an element of the remainder $B \perp \alpha$ if and only if it is a maximal subset of B that does not imply α:*

- *B' is a subset of B ($B' \subseteq B$)*
- *$\alpha \notin Cn(B')$*
- *If $B' \subset B'' \subseteq B$, then $B'' \vdash \alpha$*

An important definition is of a selection function:

Definition 2 (Selection Function) [Alchourrón et al. 1985] *Let $\mathcal{L}$ be a language and B a belief base of this language. For any sentence α, a selection function for B is a function γ such that, for any sentence $\alpha \in \mathcal{L}$:*

- *if $B \perp \alpha \neq \emptyset$, then $\gamma(B \perp \alpha) \neq \emptyset$ and $\gamma(B \perp \alpha) \subseteq B \perp \alpha$Ontology Merging using BContractor: Preliminary Report*
- *if $B \perp \alpha = \emptyset$, $\gamma(B \perp \alpha) = \{K\}$*

Informally speaking, we have the result of the contraction choosing some elements of $B \perp \alpha$ and taking their intersection. Formally:

Definition 3 (Partial Meet Contraction) [Alchourrón et al. 1985] *Let B be a belief base, α an arbitrary sentence and γ a selection function. The Partial Meet contraction function is defined as $B -_\gamma \alpha = \bigcap \gamma(B \perp \alpha)$.*

2.2. *Kernel* Contraction

This construction uses a different approach to solve the problem. Here, the construction consists in finding the minimal subsets of B that imply α and, then, remove at least one element from each of these subsets. The set of these minimal subsets is called the *kernel* of B by α, represented as $B \perp\!\!\!\perp \alpha$.

Definition 4 (Kernel Set) [Hansson 1999] *Let B be a belief base and α a sentence. A set B' is an element of the kernel $B \perp\!\!\!\perp \alpha$ if and only if it is a minimal subset of B that implies α:*

- *B' is a subset of B ($B' \subseteq B$)*
- *$\alpha \in Cn(B')$*
- *If $B'' \subset B' \subseteq B$, then $B'' \nvdash \alpha$*

An incision function selects at least one element of each kernel to be removed:

Definition 5 (Incision Function) [Hansson 1999] *Let B be a belief base. For any sentence α, an incision function for B is a function σ such that:*

- *$\sigma(B \perp\!\!\!\perp \alpha) \subseteq \bigcup(B \perp\!\!\!\perp \alpha)$ and*
- *if $\emptyset \neq X \in B \perp\!\!\!\perp \alpha$ then $X \cap \sigma(B \perp\!\!\!\perp \alpha) \neq \emptyset$*

A kernel contraction is then defined as removing from the belief base those elements from the kernels selected by the incision function:

Definition 6 [Hansson 1999] *Let B be a belief base, σ an incision function and α a sentence. The Kernel contraction function is defined as $B -_\sigma \alpha = B \backslash \sigma(B \perp\!\!\!\perp \alpha)$*

2.3. Belief Merging and Conflict Resolution

A close related area to Belief Revision is the area of Belief Merging, where instead of adding a single piece of information to a belief base, two (or more) belief bases are combined. In an operation of revision, the incoming information has higher priority over the existing belief base, while in Merging, usually the two belief bases being merged have equal priority. So the areas of Belief Revision and Merging walk hand-in-hand, sharing a large amount of activities to be carried out during the operators executions. For a review on Belief Merging, please refer to [Konieczny and Pérez 2011].

A Merging operator can also be obtained by first joining the two belief bases involved and then solving the conflicts that arise in case the bases are inconsistent. With a slight adaptation, the concepts of remainders and kernels can be used to construct Merging operators. In this paper, we will consider a construction that is based on finding the minimal inconsistent subsets of the joined bases (kernels) and removing at least one element of each.

3. Ontology Merging

When presenting our quick overview of belief revision, we did not specify the logical language used. In fact, the original paper on the AGM paradigm [Alchourrón et al. 1985] does not require a particular logic. Nevertheless, several assumptions are made on the underlying logic, such as supraclassicality, which prevents the paradigm from being directly applicable to several useful logics, such as Description Logics [Baader et al. 2003].

Ontologies describe individuals, classes, attributes of these classes and relationships between them. The OWL language[2], a W3C recommendation since 2004 for representing ontologies, is based on Description Logics - DL. DLs are subsets of First Order Logic, have a well defined semantics and are usually decidable.

[2]http://www.w3.org/TR/owl-guide/

As we can see, if we intend to work with knowledge representation, we should consider these languages for describing ontologies. Nonetheless, in many applications it is not enough to represent knowledge; we should also be able to change it and deal with its dynamics.

There have been several proposals to apply belief revision for ontologies in OWL and DL, such as [Kalyanpur 2006, Ribeiro 2013]. In this paper, we turn to the problem of applying merging operators to combine ontologies and providing an implementation based on *BContractor*.

As mentioned in the Introduction, the integration of multiple knowledge sources will, eventually, result in conflicting knowledge being joined together in a single base. This kind of problem may compromise the integrity and reliability of a knowledge base. When dealing with the integration of ontologies, it is important to distinguish inconsistency from incoherence. An ontological knowledge base is usually divided in two parts: the ABox, containing assertional knowledge about individuals, and the TBox, containing terminological knowledge about concepts and properties. An ontology is considered inconsistent if and only if there is no interpretation that could satisfy all the axioms of the base [Haase et al. 2005]. This kind of problem typically arises with assertional knowledge, i.e., the ABox. A knowledge base is considered incoherent if and only if there is a concept C such that, for all possible models for the knowledge base, C has an empty interpretation [Qi and Pan 2007]. This kind of problem typically arises with terminological knowledge, i.e., the TBox.

Several activities play important roles during the process of inconsistency solving. In [Cobe and Wassermann 2012] the authors group the most common activities developed during the resolution of a conflict into the following phases:

3.1. Kernel Building

The goal of this phase is to build minimal, conflict keeping sub-ontologies, which is closely related to the idea of kernel, i.e., S is a kernel of the inconsistent/incoherent ontology O iff: S is a subset of O, S is inconsistent/incoherent and there is no proper subset of S that is inconsistent/incoherent. We used the same designation as [Kalyanpur 2006, Wassermann 1999]. The concept of kernel is similar to the *Minimal Incoherence Preserving Sub-Ontologies* (MIPS) and *Minimally Unsatisfiability Preserving Sub-TBoxes* (MUPS) [Schlobach 2005, Haase et al. 2005]. In a typical ontology merging scenario, the user might have to examine each kernel at a time, probably using different strategies to deal with each inconsistency/incoherence.

3.2. Stratification

During this phase, the axioms in the chosen kernel are ordered according to some principle - the number of axioms that share concepts and individuals, for instance. We chose to use the same denomination presented in [Qi and Pan 2007, Meyer et al. 2005]. The goal of this phase is similar to the one of the incision functions presented earlier, which also rank axioms according to some criteria. The main difference is that stratification defines strategies to order axioms possibly from one single kernel and incision functions take as input all possible kernels, thus, we can think of incision functions as being composed by stratification strategies - responsible for ordering axioms - and a selection function - which removes the least preferred axiom from every stratified kernel.

The stratification phase can be carried out manually by domain experts [Haase and Volker 2008, Ribeiro and Wassermann 2008], or by automatic means. Now, we are going to enumerate a couple of the most common approaches for stratification.

Specific Axiom Prioritization has been proposed by Qi et al. in [Qi and Pan 2007] and its main idea, taken from [Benferhat 2003], aims to preserve the axioms that describe more general concepts, or more formally: an axiom $\phi_1 = C_1 \sqsubseteq D_1$ is more specific than the axiom $\phi_2 = C_2 \sqsubseteq D_2$ if and only if $C_1 \sqsubseteq C_2$ and $C_2 \not\sqsubseteq C_1$.

Kalyanpur, in [Kalyanpur 2006] also proposed a few algorithms for axiom ranking:

- Order by frequency: which orders by the number of kernels in which the axiom appears;
- Order by semantic relevance: which orders by the number or entailments that are lost or added if the axiom is removed; and
- Order by syntactic relevance: which orders the number of axioms that share the concepts with the axiom being ranked.

All of these strategies are also good candidates for composing incision functions. We implemented some of these approaches using the BContractor framework (see Section 4 below).

3.3. Axiom Weakening

The activities in this phase try to solve the inconsistencies (not incoherences) by modifying the axioms, weakening their restriction power. This phase is not shared with Belief Revision. When we allow the operator to weaken the formula in order to keep consistency we can no longer guarantee that the formula α will be in the resulting knowledge base. This phase is still very useful in cases when the user wants to maintain most of the information in a knowledge base and he/she does not care if the information in the knowledge base is slightly different from before the merging. The goal here is to avoid discarding whole axioms. In what follows, we list some of the main strategies found in the literature.

The first strategy we would like to point is the exception adding, described in [Qi et al. 2006]. The idea consists in transforming inconsistent kernels of the form $K = \{C \sqsubseteq D, C(a) \sqcap \neg D(a)\}$ into $K' = \{(C \sqcap \neg\{a\}) \sqsubseteq D, C(a) \sqcap \neg D(a)\}$.

In [Cobe and Wassermann 2012], the authors proposed a new way of weakening cardinality restrictions. The idea is to iteratively change the value of n in the $\leq nP$ axiom, where n is a number and P a property. They designed a weakening operator that takes all possible minimally inconsistent sets and tries to fix the largest number of inconsistencies by changing the n value. The algorithm proposed needs to check all kernel set because if the inconsistency is fixed in one specific kernel it may be the case that when the ontology is put together, the inconsistency will appear again.

The authors showed that, in this case, an approach closer to believe revision may be better suiting, where an incision function, used to choose which axioms would be weakened - instead of removed -, is composed by a single stratification strategy and is able to select the axioms involved in the conflict which may be more than one axiom from each kernel.

3.4. Axiom Removal

This phase aims to remove the axiom with the lowest priority (or trustability) in the kernel in which the user is working to solve the conflict. This approach is used in most of the works on ontology debugging [Schlobach et al. 2007, Schlobach 2005] and Belief Revision in DL [Ribeiro 2013].

4. BContractor-DL

After some decades of research and study in Belief Revision, we have many results available in the literature, including comparisons between the different possible operators. Nevertheless, there is still a gap when we consider software tools to work with these operators or computational resources analysis. There are some implementations of Belief Revision operators available. As examples, we can cite BReLS [Liberatore 1999] and Saten [Williams and Sims 2000]. However, they focus on a specific logic or construction.

Considering this scenario, the BContractor [Lundberg et al. 2012] was recently released with the purpose of being a more flexible framework for implementing and testing Belief Change operators. One of its main purposes is the possibility of extending it to implement and test operators for different kinds of logic, although so far it has been tested only for Propositional Logic, considering that most work on Belief Revision theory is based on that logic. Still, much effort is being applied to adapt such theory to other logics. Following this purpose, we describe in this section the extensions that we have made to the BContractor in order for it to support DL knowledge bases.

This implementation is restricted to the $\mathcal{SROIQ}$ DL [Horrocks et al. 2006] due to the fact that this DL family is the one that underpins OWL2, which is the language supported by most of the reasoners, including HermiT[3], the one we used. For more expressive DLs, one should use a language more expressive than OWL2 and another reasoner to support it.

The first inclusion was a new component that was needed to calculate kernels from inconsistent knowledge bases. We needed to do that because we are dealing with DL, and as presented in [Ribeiro 2013], in several DLs the negation of an axiom is not defined. So, in order to avoid the usage of negation, instead of revision we rely on the operation of semi-revision presented in [Hansson 1999] and also used in [Ribeiro 2013]. The idea is to insert α in the knowledge base and then contract it by the inconsistency.

The new component defined was a new version of the BContractor *KernelOperator*, the *KernelConflictOperator*, which is able to compute kernels from inconsistent/incoherent knowledge bases. The main difference between the two is that in *KernelConflictOperator* we are able to compute kernels from a knowledge base instead of a knowledge base plus an axiom. These two components only define interfaces and in order to use them, one need to give concrete implementations. We implemented a concrete version of the *KernelConflictOperator* in the *BlackBoxKernelOperator*, so now it is also able to use the *eval* operation on a knowledge base, instead of a pair.

After that, we have built the ground for the definition of Revision operators. We developed the *InternalKernelRevisionWithoutNegation* component, which uses an incision function and a *KernelConflictOperator*. Then it builds a new knowledge base from

[3] `http://hermit-reasoner.com/`

the execution of the incision function in the kernels built by the *KernelConflictOperator*. We use an Internal Revision technique [Ribeiro 2013] in which first we open room for the new piece of knowledge, and then we really add it. The operator behaves as follows: first it calculates the kernels for the knowledge base union α, then it executes the incision function. After that it removes the result of the incision function from the knowledge base and finally it adds α to the knowledge base.

The usage of the BContractor made it really easy to code a new revision operator as shown in Listing 1.

Listing 1. InternalKernelRevisionWithoutNegation Revision Operator

```
1 ISet<S> revise(ISet<S> base, S alpha) {
2   return base.minus(incision(kernel(base.union(alpha)))).union(alpha);
3 }
```

In Listing 1 the *kernel* operation uses the *BlackBoxKernelOperator* to build the kernel set from the union of the knowledge base with *alpha*, then the incision function takes place and calculates a cutting set - a set that contains at least one element of each kernel in the kernel set. After that we can be sure that we removed at least a single element from each kernel breaking their minimality principle, thus restoring their consistency. The result of the incision function is removed from the base with the *minus* operation and only after "making room for *alpha*" is that we include it with the *union* operation.

In addition to that, we have developed two incision functions. The first function prioritizes removing the elements that appear in the largest number of kernels. This function was described by Kalyanpur in [Kalyanpur 2006] and is very useful when we try to keep as much information as possible. So we do not remove a separate axiom from each kernel. In this way we do not remove more axioms than we need to, e.g., consider the following kernel set: $K = \{\{C \sqsubseteq D, C(a) \sqcap \neg D(a)\}, \{C \sqsubseteq D, C(b) \sqcap \neg D(b)\}\}$. By applying this incision function the resulting cutting set would be: $K' = \{C \sqsubseteq D\}$. The implemented component is called *MostFrequentFirstIncisionFunction*.

We have also developed the operator proposed by Qi et al in [Qi and Pan 2007], *MostSpecificFirstIncisionFunction*, that keeps the most general axioms in the knowledge base, e.g., consider the following kernel set: $K = \{\{C \sqsubseteq \neg D, C \sqsubseteq F, F \sqsubseteq D\}\}$. By applying this incision function the resulting cutting set would be $K' = \{C \sqsubseteq \neg D\}$[4].

The usage of these operators is simple, the user has to pass them to the *Revision Operator* being used and call the *revise* operation. In order to use incision functions in a stand-alone way, the user needs to instantiate them and pass the kernel set to their *eval* functions.

As this paper presents an ongoing work the study of the properties of the incision functions developed will be done in the future.

For the Belief Merging case, we developed a new type of operator, the *MergeOperator*, which has a single operation, *merge*, that receives two knowledge bases and produces a new one as output.

[4]Another possible cutting set would be $K'' = \{C \sqsubseteq F\}$. The BContractor-DL chooses only one of the possibilities - the first one.

We have implemented the new idea of *StratificationOperator* which aims to order the kernels according to some specific criteria and also the *WeakenOperator* that builds weaker versions of the kernels in order to restore their consistency. We then developed two stratification operators, the *FrequencyStratificationOperator* and the *GeneralityStratificationOperator* that use the same ideas from the revision incision functions: *MostFrequentFirstIncisionFunction* and *MostSpecificFirstIncisionFunction* respectively.

We have also defined a weakening operator named *NumberedRestrictionWeakenOperator* that uses the idea described in Section 3.3. When the authors presented this approach in [Cobe and Wassermann 2012] they showed that in order to use this wakening strategy we needed a incision function that selects maybe more than one single element from each kernel in the kernel set, e.g., suppose we have the following inconsistent ontology: $O = \{C \sqsubseteq \leq 1P, a_2 \neq a_3, a_2 \neq a_4, a_3 \neq a_4\, C(a_1), P(a_1, a_3), P(a_1, a_2), P(a_1, a_4)\}$. Calculating its kernel set we obtain $K = \{\{C \sqsubseteq \leq 1, C(a_1), a_2 \neq a_3, P(a_1, a_2), P(a_1, a_3)\}, \{C \sqsubseteq \leq 1, C(a_1), a_2 \neq a_4, P(a_1, a_2), P(a_1, a_4)\}, \{C \sqsubseteq \leq 1, C(a_1), a_3 \neq a_4, P(a_1, a_3), P(a_1, a_4)\}\}$. A Numbered Restriction Incision function would return the following cutting set $K' = \{\{C \sqsubseteq \leq 1, P(a_1, a_3), P(a_1, a_2), P(a_1, a_4)\}$. For this matter we developed the *NumberedRestrictionIncisionFunction*.

So, with all these, we have built the ground for the definition of the first merging operator using BContractor, which is defined as the build of a new base from the weakening of the stratified base built from the kernels of the union of the two bases being merged. Using the design ideas from the BContractor, the code for doing so is still human-readable and easy to redefine:

Listing 2. Merge Operator

```
1  ISet<S> merge(ISet<S> base1, ISet<S> base2){
2    Kernel<S> kernelSet = kernel(base1.union(base2));
3    Strata<S> stratifiedKernelSet = stratify(kernelSet);
4    ISet<S> cuttingSetToWeaken = weakenIncision(stratifiedKernelSet);
5    ISet<S> weakenedSet = weaken(cuttingSetToWeaken);
6    if(reasoner.isConsistent(base1.union(base2).minus(
7         cuttingSetToWeaken).union(weakenedSet))){
8           return base1.union(base2).minus(cuttingSetToWeaken).
9             union(weakenedSet);
10   }
11   else{
12     return base1.union(base2).minus(incision(stratifiedKernels));
13   }
14 }
```

The code listed first builds the kernelSet of the union of the two bases (line 2), after that, as explained in [Cobe and Wassermann 2012] we need to use a Most Frequent First [Kalyanpur 2006] strategy to stratify the base. This causes the $\leq nP$ axioms to appear first in the kernel inside the kernel set. This step is important so the *NumberedRestrictionIncisionFunction* knows that all kernels contain the same $\leq nP$ axiom and that that is the axiom to be weakened. After that we use the *NumberedRestrictionIncisionFunction* to select the elements from the kernel set that are going to be weakened. Although only one axiom will be weakened by the *WeakenOperator*, the *NumberedRestrictionIncisionFunction* includes the property assertions that will be used to calculate the new n value, that will be the amount of property assertions.

The *NumberedRestrictionWeakenOperator* is called by the *weaken* function, building a weakened version of the axioms selected by the *NumberedRestrictionIncisionFunction*. After that the *MergeOperator* verifies if the consistency was restored. If that is the case, then it removes the elements selected by the *NumberedRestrictionIncisionFunction* from the base and add their weakened version built by the *NumberedRestrictionWeakenOperator*.

In the worst case scenario, if that does not restore consistency, the *MergeOperator* calculates another cutting set, using a second incision function and removes those axioms from the base, restoring the consistency. The second incision function is needed because it probably will select less elements than the one used for axiom weakening.

5. Usage Example

In this section we are going to describe a small example that aims to show how the user could interact with the framework and how it can be used to restore consistency/coherence in ontologies being merged. The example is very restrict due to lack of space.

Suppose that we have the following ontologies O_1 and O_2 composed by the axioms[5]:

O_1 :
$\phi_1 = E \sqsubseteq F$,
$\phi_2 = F \sqsubseteq \leq 1P$,
$\phi_3 = E(a_1)$,
$\phi_4 = a_2 \neq a_3$,
$\phi_5 = a_2 \neq a_4$,
$\phi_6 = a_3 \neq a_4$,

O_2 :
$\phi_7 = a_2 \neq a_3$,
$\phi_8 = a_2 \neq a_4$,
$\phi_9 = a_3 \neq a_4$,
$\phi_{10} = P(a_1, a_2)$,
$\phi_{11} = P(a_1, a_3)$,
$\phi_{12} = P(a_1, a_4)$

The resulting ontology from the union of O_1 and O_2, $O = \{\phi_1, \phi_2, \phi_3, \phi_7, \phi_8, \phi_9, \phi_{10}, \phi_{11}\ \phi_{12}\}$, is inconsistent, due to the fact that the individual a_1 relates to more than 1 other individual by means of the P property, while the axiom ϕ_2 explicitly says the opposite.

We define the following approach of solving inconsistency after joining the ontologies: we will define a new *MergeOperator* that uses a *BlackBoxKernelOperator*, a *FrequencyStratificationOperator* for stratification, a *NumberedRestrictionIncisionFunction* for selecting axioms to be weakened and a *NumberedRestrictionWeakenOperator* as a weakening operator.

The kernel building operator produces the following kernels, $K_1 = \{\phi_1, \phi_2, \phi_7, \phi_3, \phi_{10}, \phi_{11}\}$, $K_2 = \{\phi_1, \phi_2, \phi_8, \phi_3, \phi_{10}, \phi_{12}\}$ and $K_3 = \{\phi_1, \phi_2, \phi_9, \phi_3, \phi_{11}, \phi_{12}\}$.

When the *FrequencyStratificationOperator* is executed on the kernels obtained previously, it results in the following stratified kernels $K'_1 = \{\phi_1, \phi_2, \phi_3, \phi_{10}, \phi_{11}, \phi_7\}$, $K'_2 = \{\phi_1, \phi_2, \phi_3, \phi_{10}, \phi_{12}, \phi_8\}$ and $K'_3 = \{\phi_1, \phi_2, \phi_3, \phi_{11}, \phi_{12}, \phi_9\}$. The calculated frequency for axioms was 3 for ϕ_1, ϕ_2, ϕ_3, 2 for ϕ_{10}, ϕ_{11}, ϕ_{12} and 1 for ϕ_7, ϕ_8, ϕ_9.

[5]Note that axioms ϕ_4,ϕ_5, ϕ_6 are the same as ϕ_7, ϕ_8, ϕ_9.

The *NumberedRestrictionIncisionFunction* then calculates the cutting set $CS = \{\phi_1, \phi_{10}, \phi_{11}, \text{and } \phi_{12}\}$ that can be used to feed the *NumberedRestrictionWeakenOperator*. The weakening operator execution results in the modification of the axiom ϕ_2 into the axiom $\phi_2' = F \sqsubseteq \leq 3P$ that is used to update the union of the O_1 and O_2 ontologies. Just to give an idea of the syntax, the output of the weakening process is at follows:

```
ClassAssertion(<#E> <#a1>)
SubClassOf(<#F> ObjectMaxCardinality(3 <#P> owl:Thing))
SubClassOf(<#E> <#F>)
DifferentIndividuals(<#a2> <#a3> )
DifferentIndividuals(<#a3> <#a4> )
DifferentIndividuals(<#a2> <#a4> )
ObjectPropertyAssertion(<#P> <#a1> <#a4>)
ObjectPropertyAssertion(<#P> <#a1> <#a3>)
ObjectPropertyAssertion(<#P> <#a1> <#a2>)
```

6. Conclusion

In this paper we showed an extension of the BContractor framework in order to: (a) apply it to Description Logics and (b) implement Merging operators. The extension shows that BContractor is indeed independent of the underlying logics and that it is easily extensible to implement different Belief Change operators, as promised on its release. It permits re-usability of code and more modularized and organized software applications.

The code for the extension of BContractor is freely available at `http://www.ime.usp.br/~rmcobe/OntologyMerging/`. We have also integrated the extension with the Protégé[6] revision plugin first described in [Ribeiro and Wassermann 2008] and available at `https://code.google.com/p/review-and-contract/`.

Future work includes the implementation of the Ontology Merging operators as parts of the Protégé plug-in and empirically testing the different merging strategies on benchmark ontologies. We also plan to study the formal properties of the incision functions implemented and described in Section 4.

Acknowledgments The first and the second authors are supported by the São Paulo Research Foundation (FAPESP), grant numbers 2008/10498-8 and 2011/04477-0, respectively. The third author is partially supported by CNPq, grant number 304043/2010-9. This research is part of FAPESP project OnAIR 2010/19111-9.

References

Alchourrón, C., Gardenfors, P., and Makinson, D. (1985). On the logic of theory change. *Journal of Symbolic Logic*, 50(02):510–530.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., editors (2003). *The Description Logic Handbook*. Cambridge University Press.

Benferhat, S. (2003). A stratification-based approach for handling conflicts in access control. In *SACMAT'03*, pages 189–195.

Cobe, R. and Wassermann, R. (2012). Ontology merging and conflict resolution. In *Workshop on Belief Change, Non-monotonic Reasoning and Conflict Resolution (BNC)*.

[6]Protégé is a free and open-source ontology editor, serving as a framework for knowledge bases. It was developed by Stanford University, also receiving collaboration from the University of Manchester. Available at `http://protege.stanford.edu/`

Gärdenfors, P. (1988). *Knowledge in Flux - Modeling the Dynamics of Epstemic States*. MIT Press.

Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., and Sure, Y. (2005). A framework for handling inconsistency in changing ontologies. In *ISWC' 05*, pages 353–367. Springer.

Haase, P. and Volker, J. (2008). Ontology learning and reasoning — dealing with uncertainty and inconsistency. In *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNCS*, pages 366–384. Springer.

Hansson, S. O. (1991). *Belief Base Dynamics*. PhD thesis, Uppsala University, Suécia.

Hansson, S. O. (1999). *A Textbook of Belief Dynamics*. Kluwer Academic Publishers, Norwell, MA, USA.

Horrocks, I., Kutz, O., and Sattler, U. (2006). The even more irresistible sroiq. In *KR*, pages 57–67.

Kalyanpur, A. (2006). *Debugging and repair of owl ontologies*. PhD thesis, University of Maryland, College Park, MD, USA.

Konieczny, S. and Pérez, R. P. (2011). Logic based merging. *Journal of Philosophical Logic*, 40(2):239–270.

Levi, I. (1977). Subjunctives, dispositions and chances. *Synthese*, 34(4):423–455.

Liberatore, P. (1999). BReLS: a system for revising, updating, and merging knowledge bases. In *Proceedings of NRAC*.

Lundberg, R., Ribeiro, M., and Wassermann, R. (2012). A framework for empirical evaluation of belief change operators. In *SBIA 2012*, LNAI 7589, pages 12–21. Springer.

Meyer, T., Lee, K., and Booth, R. (2005). Knowledge integration for description logics. In *AAAI'05*, pages 645–650.

Qi, G., Liu, W., and Bell, D. (2006). A revision-based approach to handling inconsistency in description logics. *Artif. Intell. Rev.*, 26:115–128.

Qi, G. and Pan, J. (2007). A stratification-based approach for inconsistency handling in description logics. In *IWOD'07*, page 83, Innsbruck.

Ribeiro, M. and Wassermann, R. (2008). The ontology revisor plug-in for Protégé. In *WONTO*.

Ribeiro, M. M. (2013). *Belief Revision in Non-Classical Logics*, volume XI of *Springerbriefs in Computer Science*. Springer.

Schlobach, S. (2005). Debugging and semantic clarification by pinpointing. In *The Semantic Web: Research and Applications*, LNCS, pages 27–44. Springer.

Schlobach, S., Huang, Z., Cornet, R., and van Harmelen, F. (2007). Debugging incoherent terminologies. *Journal of Automated Reasoning*, 39:317–349.

Wassermann, R. (1999). *Resource-Bounded Belief Revision*. PhD thesis, Universiteit van Amsterdam.

Williams, M.-A. and Sims, A. (2000). Saten: An object-oriented web-based revision and extraction engine. *CoRR*, cs.AI/0003059.

Assertion Role in a Hybrid Link Prediction Approach through Probabilistic Ontology

Marcius Armada[1] **, Kate Revoredo**[1] **, José Eduardo Ochoa Luna**[2] **, Fabio Gagliardi Cozman** [3]

[1] Departamento de Informática Aplicada, Unirio
Av. Pasteur, 458, Rio de Janeiro, RJ, Brazil

[2]Universidad Católica San Pablo
Quinta Vivanco s/n , Urb. Campiña Paisajista, Arequipa, Perú

[3]Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Morais 2231, São Paulo - SP, Brazil

{marcius.oliveira, katerevoredo}@uniriotec.br, eduardo.ol@gmail.com, fgcozman@usp.br

Abstract. *Link prediction in a network is mostly based on information about the neighborhood topology of the nodes. Recently, the interest for hybrid link prediction approaches that combine topology information with information about the network individuals, has grown. However, considering the whole set of individuals may not be necessary and sometimes not even suitable. Therefore, mechanisms to automatically discover the relevant set of individuals are demanding. In this paper, we encompass this problem by proposing an algorithm that combines structure and semantic metrics to find the set of relevant individuals. We empirically evaluate this proposal analyzing the assertion role of these individuals when predicting a link through a probabilistic ontology.*

1. Introduction

Many social, biological, and information systems can be well described by networks, where nodes represent objects (individuals), and links denote the relations or interactions between nodes. These networks have a dynamic behavior, thus nodes and links can appear and disappear rapidly. In this scenario, predicting a possible link in a network, this is predicting a future occurrence of a not yet existing relationship, is an interesting issue that has received significant attention. For instance, one may be interested on finding potential friendship between two persons in a social network, or a potential collaboration between two researchers. In short, *link prediction* aims at predicting whether two nodes should be connected given previous information about their relationships or interests.

Hasan and Zaki [Al Hasan and Zaki 2011] survey representative link prediction methods, classifying them in three groups. In the first group, feature-based methods construct pairwise features to use in classification. The majority of the features are extracted from the graph topology by computing similarity based on the neighborhood of the pair of nodes, or based on ensembles of paths between the pair of nodes [Liben-Nowell and Kleinberg 2007]. Semantic information has also been used as features [Sachan and Ichise 2011, Wohlfarth and Ichise 2008]. The second group includes probabilistic approaches that model the joint probability for entities in a network by Bayesian graphical models [Wang et al. 2007]. The third group employs linear algebraic

approaches that compute the similarity between nodes in a network by rank-reduced similarity matrices [Kunegis and Lommatzsch 2009].

In [Ochoa-Luna et al. 2013], an approach for link prediction that combines Bayesian graphical models and semantic-based features was proposed. To represent semantic-based features, a probabilistic ontology represented with the probabilistic description logic called Credal $\mathcal{ALC}$ (CR$\mathcal{ALC}$) [Cozman and Polastro 2009] was used. This probabilistic description logic extends the popular logic $\mathcal{ALC}$ [Schmidt-Schauß and Smolka 1991] with *probabilistic inclusions*. These are sentences, such as $P(\mathsf{Professor}|\mathsf{Researcher}) = 0.4$, specifying the probability that an element of the domain is a Professor given that it is a Researcher. Exact and approximate inference algorithms for CR$\mathcal{ALC}$ have been proposed [Cozman and Polastro 2009], using ideas inherited from the theory of Relational Bayesian Networks [Jaeger 2002].

When using semantic features, information about the individuals of the domain are considered. However, information about all individuals may not be necessary and sometimes not even suitable. Therefore, mechanisms that automatically select the relevant individuals are important. In [Ochoa-Luna et al. 2013], a first discussion about this matter was done, where structure features were considered to select the most relevant individuals. In this paper, we extend this idea and evaluate alternative methods for selecting the set of relevant individuals. We empirically evaluate our proposal using a probabilistic ontology, represented in CR$\mathcal{ALC}$, for modeling the domain.

The paper is organized as follows. Section 2 reviews basic concepts of probabilistic description logics and link prediction. Our proposal for selecting the most relevant individuals related to the two being analyzed for link prediction is presented in Section 3. Section 4 describes experiments, and Section 5 concludes the paper and discusses some future work.

2. Background

This section briefly review probabilistic description logics and link prediction methods, with a focus on concepts and techniques that are later used.

2.1. Probabilistic Description Logics and CR$\mathcal{ALC}$

Description logics (DLs) form a family of representation languages that are typically decidable fragments of first order logic (FOL) [Baader and Nutt 2002]. Knowledge is expressed in terms of *individuals*, *concepts*, and *roles*. The semantics of a description is given by a *domain* $\mathcal{D}$ (a set) and an *interpretation* $\cdot^{\mathcal{I}}$ (a functor). Individuals represent objects through names from a set $N_{\mathsf{I}} = \{a, b, \ldots\}$. Each *concept* in the set $N_{\mathsf{C}} = \{C, D, \ldots\}$ is interpreted as a subset of a domain $\mathcal{D}$. Each *role* in the set $N_{\mathsf{R}} = \{r, s, \ldots\}$ is interpreted as a binary relation on the domain. An assertion states that an individual belongs to a concept of that a pair of individuals satisfies a role. An *ABox* is a set of assertions.

A popular description logic is $\mathcal{ALC}$ [Schmidt-Schauß and Smolka 1991]; given its importance to our proposal, we briefly review it here. Constructors in $\mathcal{ALC}$ are *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *existential* restriction ($\exists r.C$), and *value* restriction ($\forall r.C$). Concept *inclusions* and *definitions* are denoted respectively by $C \sqsubseteq D$ and $C \equiv D$, where C and D are concepts. Concept $C \sqcup \neg C$ is denoted by $\top$, and concept $C \sqcap \neg C$ is denoted by $\bot$. The semantics of these constructs is given by a domain

$\mathcal{D}$ and an *interpretation* $\mathcal{I}$ as follows: each individual a is mapped into an element $a^{\mathcal{I}}$; each concept C is mapped into a subset $C^{\mathcal{I}}$ of the domain; each role r is mapped into a binary relation $r^{\mathcal{I}}$ in the domain; moreover,

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$;
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$;
- $(\neg C)^{\mathcal{I}} = \mathcal{D} \backslash C^{\mathcal{I}}$;
- $(\exists r.C)^{\mathcal{I}} = \{x \in \mathcal{D} | \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$;
- $(\forall r.C)^{\mathcal{I}} = \{x \in \mathcal{D} | \forall y : (x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$.

Finally, $C \sqsubseteq D$ is interpreted as $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and $C \equiv D$ is interpreted as $C^{\mathcal{I}} = D^{\mathcal{I}}$.

An example may be useful. Consider the following concept definition:

$$\mathsf{Researcher} \equiv \mathsf{Person} \ \sqcap \ \exists \mathsf{hasPublication.BibItem} \tag{1}$$

specifying that researchers are individuals who are persons and who have published a bibliographic item.

Several *probabilistic* description logics have appeared in the literature [Lukasiewicz and Straccia 2008, Klinov 2008]. An example is the probabilistic description logic CR$\mathcal{ALC}$, which is a probabilistic extension of the description logic $\mathcal{ALC}$. It keeps all constructors of $\mathcal{ALC}$, but only allows concept names on the left hand side of inclusions/definitions. Additionally, in CR$\mathcal{ALC}$ one can have probabilistic inclusions such as $P(C|D) = \alpha$ or $P(r) = \beta$ for concepts C and D, and for role r (in this paper we only consider equality in probabilistic inclusions/definitions). If the interpretation of D is the whole domain, then we simply write $P(C) = \alpha$. The semantics of these inclusions is roughly (a formal definition can be found in Ref. [Cozman and Polastro 2009]) given by:

$$\forall x \in \mathcal{D} \ : \ P(C(x)|D(x)) = \alpha,$$

$$\forall x \in \mathcal{D}, y \in \mathcal{D} \ : \ P(r(x, y)) = \beta.$$

We assume that every terminology is acyclic: no concept uses itself (where "use" is the transitive closure of "directly use"; we say that C directly uses D if D appears in the right hand side of an inclusion/definition, or in the conditioning side of a probabilistic inclusion). This assumption allows one to represent any terminology $\mathcal{T}$ through a directed acyclic graph. Such a graph, denoted by $\mathcal{G}(\mathcal{T})$, has each concept name and role name as a node, and if a concept C directly uses concept D, that is if C and D appear respectively in the left and right hand sides of an inclusion/definition, then D is a *parent* of C in $\mathcal{G}(\mathcal{T})$. Each existential restriction $\exists r.C$ and each value restriction $\forall r.C$ is added to the graph $\mathcal{G}(\mathcal{T})$ as a node, with an edge from r and C to each restriction directly using it. Each restriction node is a *deterministic* node in that its value is completely determined by its parents.

Consider, as an example, a terminology $\mathcal{T}_R$ containing the sentence in Expression (1), plus $P(\mathsf{Person}) = 0.2$, $P(\mathsf{BibItem}) = 0.6$, $P(\mathsf{hasPublication}) = 0.1$; its graph is depicted in the left of Figure 1.

The semantics of CR$\mathcal{ALC}$ is based on probability measures over the space of interpretations, for a fixed domain. To make sure a terminology specifies a single probability measure, a number of additional assumptions are adopted: the domain is assumed

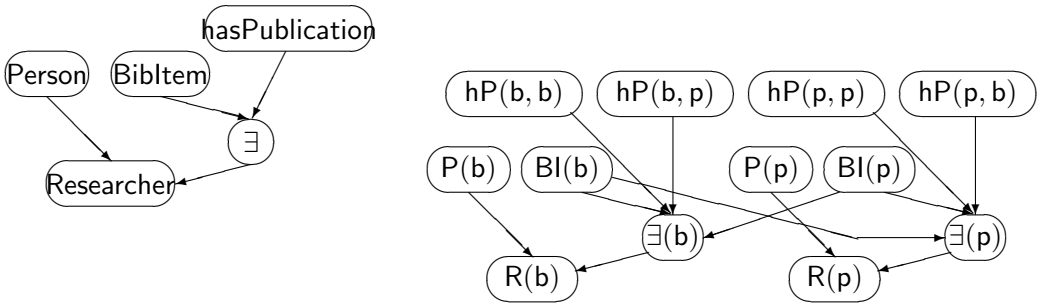


Figure 1. Graph $\mathcal{G}(\mathcal{T}_R)$ (left Figure) and Bayesian network over indicator functions of assertions, produced by grounding the terminology $\mathcal{T}_R$ (right figure)

finite, fixed, and known; the unique-name assumption and the rigidity assumption for individuals (as usual in first-order probabilistic logic [Fagin et al. 1990]) are assumed; a single concept name appears in the left hand side of any inclusion or definition and in the conditioned side of any probabilistic inclusion; and finally a Markov condition imposes independence of any grounding of concept/role conditional on the groundings of its corresponding parents in the graph $\mathcal{G}(\mathcal{T})$ [Cozman and Polastro 2009]. Given these assumptions, a set of sentences $\mathcal{T}$ in CR$\mathcal{ALC}$ defines a *relational Bayesian network* [Jaeger 2002] whose underlying graph is exactly $\mathcal{G}(\mathcal{T})$.

Considering the domain $\mathcal{D} = \{\mathsf{bob}, \mathsf{paper}\}$ and the set of assertions $\mathcal{A} = \{$ $\mathsf{Person}(\mathsf{bob}), \mathsf{Researcher}(\mathsf{bob}), \mathsf{BibItem}(\mathsf{paper}), \mathsf{hasPublication}(\mathsf{bob}, \mathsf{paper})$ $\}$, inferences such as $P(\mathsf{A}_o(\mathsf{a}_0)|\mathcal{A})$ can be computed by grounding the terminology, where grounding means that all existing variables must be replaced by constants. In our case they are replaced by the individuals in the domain and the grounding process generates a "slice" for each individual. The right Bayesian network in Figure 1 shows a grounding for terminology $\mathcal{T}$ where two slices, one for individual bob and another for individual paper, are built (for the sake of space, names are abbreviated). At first sight the resulting Bayesian network may seem odd, with nodes like $\mathsf{Bibitem}(\mathsf{bob})$ or $\mathsf{Person}(\mathsf{paper})$, but since we are not based on the "closed world" assumption then anything we not currently known can be either true or false. For large domains, exact probabilistic inference is in general quite hard due to the complexity of the resulting grounded Bayesian network but variational algorithms that approximate such probabilities are available in the literature [Cozman and Polastro 2009] in an attempt to deal whith this problem.

2.2. Link Prediction

The task we are interested in can be defined as follows [Liben-Nowell and Kleinberg 2007]. One is given a network (a graph) G consisting of a set of nodes V (represented by letters a, b, etc) and a set of edges E, where an edge represents an interaction between nodes. Interactions may be tagged with times, and the link prediction problem may be one of predicting the existence of edges in a time interval, given the edges observed in another time interval. Here we are interested in a static problem where we are given nodes and edges, except for the edge between two nodes a and b, and we must then predict whether there is an edge between a and b.

Many different tools are used for link prediction, some of which, like matrix factorization, are related to the massive size of datasets; other tools are directly related to the

existence of links between nodes. One can use classifiers that, based on network features and measures, classify each tentative link as existing or not [Al Hasan and Zaki 2011]; one may also resort to collective classification over the whole set of possible links [Getoor and Diehl 2005]. Several such techniques are based on computing measures of proximity/similarity between nodes in a network [Liben-Nowell and Kleinberg 2007, Lü and Zhou 2011].

Other approaches consider semantic features. The degree of semantic similarity among entities can be useful to predict links that might be missed by simple topological or frequency-based features [Wang et al. 2007]. One way of capturing semantic similarity is by considering documents related to nodes in the network. A simple example of semantic similarity is the keyword match count between two authors [Hasan et al. 2006]. A more sophisticated method makes use of the well-known techniques such as TFIDF feature vector representation and the cosine measure to compute similarity [Wang et al. 2007]. The latter measure, for documents d_1 and d_2, is obtained by creating vector representations $\overrightarrow{V}(d_1)$ and $\overrightarrow{V}(d_2)$ that countain word counts weighted by their TFIDF (Term Frequency - Inverse Document Frequency) measures. The similarity measure is then

$$\text{cosine}(d_1, d_2) = \frac{\overrightarrow{V}(d_1) \cdot \overrightarrow{V}(d_2)}{|\overrightarrow{V}(d_1)||\overrightarrow{V}(d_2)|},$$

where the dot product is used in the numerator and the Euclidean distance is used in the denominator. To recall, the TFIDF weighting scheme assigns to term t a weight in document d given by $\text{TFIDF}_{t,d} = \text{TF}_{t,d} \times \text{IDF}_t$, where $\text{TF}_{t,d}$ is the term frequency in d, and IDF_t is the inverse document frequency of t, given by $\text{IDF}_t = \log \frac{\text{N}}{\text{DF}_t}$, for N the total number of documents and DF_t the number of documents containing the term.

Approaches to link prediction can be understood not only by considering the kinds of tools employed, but also by examining the model that is used to represent the network as a whole. Typically one assumes some sort of probabilistic mechanism that at least partially explains the existence of edges, perhaps together with domain-specific knowledge (for instance, domain theories about human relationships) [Goldenberg et al. 2010, Newman 2003]. Thus the simplest network model is the Erdös-Rènyi random graph: each pair of nodes can be connected with identical probability. More sophisticated models resort to hierarchical specification of link probabilities, or to grouping of nodes within blocks of varying probability.

One way to capture the probabilistic structure of a network is through graph-based models such as Markov random fields or Bayesian networks [Pearl 1988]. However, these languages are well suited to express independence relations between a fixed set of random variables; when nodes and links are to be dealt within graphs, it is best to consider modeling languages that can specify Markov random fields and Bayesian networks over relational structures. Indeed many proposals for link prediction resort to such languages, from seminal work by Getoor et al [Getoor et al. 2002] and Taskar et al [Taskar et al. 2003]. The presence of relational structure lets one to represent properties of individuals nodes, of links, of communities; one can then compute the probability of specific links, and estimate such probabilities from data.

In [Ochoa-Luna et al. 2013], this modeling strategy was followed using the probabilistic description logics CR$\mathcal{ALC}$. The interest in models based on description log-

ics is justified given recent results on the importance of ontologies in organizing information that can be used in link prediction [Aljandal et al. 2009, Caragea et al. 2009, Thor et al. 2011]. While other link prediction implementation usually focus in one kind of feature, the one using CR$\mathcal{ALC}$ showed to be able to mix different features such as semantic, numeric and topological. Being a versatile solution doesn't make it easier to be modeled than other solutions, but as a novel approach there is still room for evolution and further experimentation.

3. Assertion Role in Link Prediction through a Probabilistic Ontology

Given a network (a graph) G consisting of a set of nodes V and a set of edges E, where an edge represents an interaction between nodes. For a link prediction task considering semantic features, we follow the approach proposed in [Ochoa-Luna et al. 2013] and model the domain using a probabilistic ontology (O) represented in CR$\mathcal{ALC}$. Nodes in G are individuals of a concept C in O and edges are instances of a role R in O. Thus, the network G is built encompassing assertions about concept C and role R. For instance, in a co-authorship network, assertions for concept Researcher are represented by nodes and assertions for role sharePublication are represented by relationships between two nodes. Figure 2 depicts a network for the assertions shown in Figure 3.

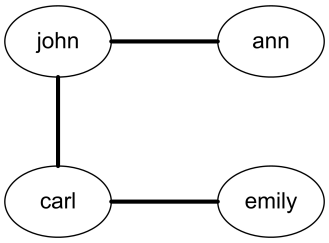


Figure 2. Network encompassing assertions of the ABox in Figure 3.

Researcher(john). Researcher(ann). Researcher(carl).
Researcher(emily). sharePublication(john, ann).
sharePublication(john, carl). sharePublication(carl, emily).

Figure 3. Example of an ontology ABox.

The probabilistic ontology O can model the domain widely, thus having other concepts and roles beyond the ones encompassing the network. For instance, an ontology describing the co-authorship domain is shown in Figure 4.

TBox:

P(Publication)	=	0.3
P(sharePublication)	=	0.22
P(hasSameInstitution)	=	0.14
Researcher	≡	Person ⊓ ∃hasPublication.BibItem
P(PublicationCollaborator	|	Researcher ⊓ ∃sharePublication.Researcher) = 0.91

ABox:

Researcher(john). Researcher(ann). Researcher(carl).
Researcher(emily). sharePublication(john, ann).
sharePublication(john, carl). sharePublication(carl, emily).
Publication(p1). Publication(p2)

Figure 4. A probabilistic ontology for the co-authorship domain.

Predicting a link between two nodes a and b in a network G concerns evaluating whether an edge between a and b should be included. In the semantic link prediction task,

where the domain is modeled through CR$\mathcal{ALC}$, the problem can be rewritten as evaluating if the considered role between individuals a and b may exist in a given ontology. Thus, the semantic link prediction task considered in this paper can be described as: compute the probability of an assertion concerning the role that provides the semantic of relationships in the network G given an ABox of asserted concepts and roles of the domain.

Because domain knowledge is expressed with CR$\mathcal{ALC}$, questions about probability of assertions can be answered by inference in CR$\mathcal{ALC}$. For instance, the question "what is the probability of Emily and Ann share a publication given some information about the domain?" can be translated into $P(\mathsf{sharePublication}(\mathsf{emily}, \mathsf{ann})|\mathcal{A})$, where $\mathcal{A}$ represents the ABox with assertions about the domain. If this probability is higher than a suitable threshold then the assertion may be considered true and a link introduced in G.

Intuitively, the inference quality of any assertion's probability rests in the used assertions contained in $\mathcal{A}$. While one can suppose that more assertions leads to more accurate calculated probabilities, this is not always true. Some individuals may not be related to the ones being analyzed and therefore their assertions may not impact the evaluation. Thus it is unnecessary to consider evidence (assertion) about them. Moreover, in some case may even be impractical to reason about all individuals of the domain due to limits in computational resources or long response times. Hence it is important to filter out assertions and to focus on the most relevant ones.

We are interested in predicting a relationship between two specific nodes, a and b. Therefore, we argue that assertions directly related to these two individuals, and to other individuals strongly related to them in the network, are more relevant for link prediction than assertions on other individuals in the network. The link prediction algorithm (see Algorithm 1) will not only be scalable but will be more accurate if we only consider assertions about a, b and the individuals strongly related to them in our inferences. To do so, we must specify the set $\mathcal{A}(a, b)$ of elements of the domain that are deemed strongly related to a and b.

Algorithm 1: Algorithm for link prediction (adapted from [Ochoa-Luna et al. 2013]).

Require: a network G, an ontology $\mathcal{O}$, a role $\hat{r}$ representing links in the network, a concept $\hat{C}$ specifying the nodes in the network and a threshold γ.
Ensure: a set of predicted links L

```
initialize L = ∅;
for all pair of instances (a, b) of nodes in G do
  if there is no link between nodes a and b in G then
    find A(a, b);
    E=assertions about A(a, b);
    infer probability P(r(a, b)|E) using the relational Bayesian network created
    from the ontology O;
    if P(r(a, b)|E) > γ then
      add link between a and b to L.
    end if
  end if
end for
```

In [Ochoa-Luna et al. 2013] the strategy adopted to define $\mathcal{A}(a, b)$ was to consider nodes along paths between a and b. In this paper, we argue that not only structural metrics can define the best set $\mathcal{A}(a, b)$ and we evaluate the performance of structural and semantic approaches for selecting the most relevant individuals for a link prediction task. The following approaches were considered:

i) $\mathcal{A}(a, b) = \mathcal{A}_{adj}(a, b)$, where $\mathcal{A}_{adj}(a, b) = adjacent(a) \cup adjacent(b)$. Defines $\mathcal{A}(a, b)$ as the set of nodes adjacent to a union the set of nodes adjacent to b.
ii) $\mathcal{A}(a, b) = \mathcal{A}_{Padj}(a, b)$, where $\mathcal{A}_{Padj}(a, b) = A_0(a, b) \cup_{i \in \mathcal{A}_0(a,b)} adjacent(i)$ and $\mathcal{A}_0(a, b) = \{a\} \cup \{b\} \cup path(a, b)$. Defines $\mathcal{A}(a, b)$ as the set of all nodes in the path between a and b union their adjacent nodes and the adjacents of a and b.
iii) $\mathcal{A}(a, b) = f_{semantic}(\mathcal{A}_{adj}(a, b))$. Defines $\mathcal{A}(a, b)$ as the set of nodes contained in $\mathcal{A}_{adj}(a, b)$ that are most semantically related to a and b considering a semantic function $f_{semantic}$.
iv) $\mathcal{A}(a, b) = f_{semantic}(\mathcal{A}_{Padj}(a, b))$. Defines $\mathcal{A}(a, b)$ as the set of nodes contained in $\mathcal{A}_{Padj}(a, b)$ that are most semantically related to a and b considering a semantic function $f_{semantic}$.

An experimental evaluation was conducted and will be described in the next section to evaluate the benefits of these metrics. Moreover, a discussion around the role of the assertions about individuals for the semantic link prediction task is also presented.

4. Experiments

Experiments have been conducted to evaluate the benefits of considering structural and semantic metrics for selecting the most relevant individuals for the semantic link prediction task. A real world data repository, the Lattes curriculum platform, was used. This section reports the steps involved in this process and the results found.

4.1. Scenario Description

The Lattes platform is the public repository of brazilian scientific curricula that consists of approximately a million registered documents. Information is encoded in HTML format, ranging from personal information such as name and professional address to publication lists, administrative tasks, research areas, research projects and advising/advisor information. There is implicit relational information in these web pages, for instance collaboration networks are built by advising/adviser links, shared publications, and so on.

To perform experiments we have randomly selected eight thousand researchers and their relationships from the Lattes platform. Assertions were extracted concerning these researchers. For instance, if a parser finds that a researcher John has two publications (p_1, p_2) and a researcher Ann has two (p_2, p_3), where p_2 was done in collaboration with John, then assertions, as the following, are extracted:

$\mathsf{Researcher(john)}$, $\mathsf{Researcher(ann)}$,
$\mathsf{Publication(p_1)}$, $\mathsf{Publication(p_2)}$, $\mathsf{Publication(p_3)}$,
$\mathsf{hasPublication(john, p_1)}$, $\mathsf{hasPublication(john, p_2)}$,
$\mathsf{hasPublication(ann, p_2)}$, $\mathsf{hasPublication(ann, p_3)}$
$\mathsf{sharePublication(john, ann)}$.

A probabilistic ontology was then learned using algorithms in the literature [Ochoa-Luna et al. 2011, Revoredo et al. 2010]. This ontology is comprised by 24 probabilistic inclusions and 17 concept definitions.

The concept Researcher indicates whether an element of the domain is a node in the network (hence for each assertion of concept Researcher a node exists in the network) and the role sharePublication indicates whether a pair of elements of the domain are linked in the network (hence for each assertion of role sharePublication a link exists in the network). Using this data, link probabilities were computed through inference in the CR$\mathcal{ALC}$ ontology.

4.2. Methodology

In this section, we describe our main design choices to run experiments. Given the 8000 selected researchers, there exist 31996000 possible link relationships. To perform link prediction we have considered collaborations based on co-authorship on publications (there are 2837206 publications). After analysing these publications we identified 95011 true positive links among researchers based on co-authorship. From the available data we randomly selected links so that the used dataset in the experiments was comprised by 1000 positive links and 1000 negative links (balanced datasets).

Although we can use probabilistic inference to decide whether there is a link between two nodes, to perform comparisons among the structural and semantic metrics described in Section 3 we resort to a classification algorithm approach through the Logistic regression algorithm.

Beyond the 4 metrics described in Section 3 we also considered:

v) the metric proposed in [Ochoa-Luna et al. 2013]: $\mathcal{A}(a,b) = \mathcal{A}_{path}(a,b)$, where $\mathcal{A}_{path}(a,b)$ defines the set of nodes in the paths between a and b .
vi) $\mathcal{A}(a,b) =$ random selection of 10 nodes in the network.

The metric v will permit us compare our proposal with the previous one presented in [Ochoa-Luna et al. 2013]. For this metric, since computing all paths (∞) is expensive, we follow Ochoa et al. and only considered paths of length at most four ($i \leq 4$).

The semantic feature we considered was keyword match. For each researcher a document with the words appearing in the title of his publications (removing stop words) is considered. Thus, a researcher is represented as a set of words, which allows us to compute a semantic feature: the keyword *match* count between two researchers [Hasan et al. 2006]. Using this feature we were able to select the top 10 researchers with the most words in common with a an b.

Finally, the probability $P(r(x,y)|E)$, given by our probabilistic description logic model, is used as a numerical feature in the classification model, in order to investigate whether it can improve the classification approach for link prediction.

4.3. Results

In order to evaluate suitability of our approach in predicting co-authorships in the Lattes dataset, several experiments were conducted. Each metric, through the probabilistic logic scores found, has been considered as isolated features in our clasification algorithm. After

Table 1. Classification results for dataset Lattes on accuracy (%) for baseline features used for selecting individuals used for generating assertions for inference in CR$\mathcal{ALC}$: metric i, metric ii, metric iii, metric iv, metric v, metric vi.

Feature	Lattes (acc.)	Avg(#) of selected individuals
CR$\mathcal{ALC}$ + metric i	99.93%	501
CR$\mathcal{ALC}$ + metric ii	99.86%	545
CR$\mathcal{ALC}$ + metric iii	99.88%	10
CR$\mathcal{ALC}$ + metric iv	99.65%	10
CR$\mathcal{ALC}$ + metric v	92.41%	26
CR$\mathcal{ALC}$ + metric vi	71.14%	10

a ten-fold cross validation process, the classification algorithm yielded results on accuracy for the dataset which are depicted in Table 1.

The results shows us that randomly selecting individuals for assertion generation (metric vi) obtained the worse accuracy in comparison to the other metrics with only 71% while all the other obtained accuracies greater than 90%. Thus, it is important to use the best possible assertions in the inference.

All other results show little differences in accuracy between each other but those metrics which don't use the semantic feature (metric i and ii) needed about 50 times more individuals to obtain near the same results. This demonstrates that the quality of the selected individuals, using the semantic feature, and the assertions generated from them were able to keep the CR$\mathcal{ALC}$ link prediction algorithm scalable and the quality of the predictions high.

5. Conclusion

In this paper, we have evaluated the role of assertions about individuals for the semantic link prediction task. We follow the approach introduced in [Ochoa-Luna et al. 2013] and considered a probabilistic ontology, represented with the probabilistic description logic CR$\mathcal{ALC}$, for modeling the domain. Thus, given a collaborative network, interests and graph features are encoded through the probabilistic ontology.

To predict links, probabilistic inference is used. Structural and semantic metrics are combined in order to select the most relevant individuals for the prediction link task. Therefore, only the necessary individuals are used and results have shown the importance of selecting the best individuals from the avaiable ones. Moreover, this approach makes the proposal scalable. Our proposal was evaluated on an academic domain, where links among researchers were predicted and was able to attain accuracies greater than 90% as shown in Table 1.

Compared to previous work, our approach employs a rich ontology (as opposed to simple is-a terminologies) that can encode substantial information about the domain. Hierarchical structure can be encoded together with knowledge about specific nodes in a network — we plan to explore richer ontologies in the future. Our proposal attains better scalability than previous proposals that have tried to explore probabilistic relational models for similar purposes but we plan to experiment with other new and state-of-the-

art selection algorithms in the search for the best set of assertions to be used in the link prediction task.

6. Acknowledgment

This work is being accomplished in the context of the "Infrastructure for the Management of Scientific Experiments in Computational Modeling" project, granted by CNPq, No. 559998/2010-4.

References

Al Hasan, M. and Zaki, M. J. (2011). A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer.

Aljandal, W., Bahirwani, V., Caragea, D., and Hsu, H. (2009). Ontology-aware classification and association rule mining for interest and link prediction in social networks. In *AAAI 2009 Spring Symposium on Social Semantic Web: Where Web 2.0 Meets Web 3.0*, Standford,CA.

Baader, F. and Nutt, W. (2002). Basic description logics. In *Description Logic Handbook*, pages 47–100. Cambridge University Press.

Caragea, D., Bahirwani, V., Aljandal, W., and Hsu, W. H. (2009). Ontology-based link prediction in the livejournal social network. In *SARA'09*, pages 1–1.

Cozman, F. G. and Polastro, R. B. (2009). Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Conference on Uncertainty in Artificial Intelligence*.

Fagin, R., Halpern, J. Y., and Megiddo, N. (1990). A logic for reasoning about probabilities. *Information and Computation*, 87:78–128.

Getoor, L. and Diehl, C. P. (2005). Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12.

Getoor, L., Friedman, N., Koller, D., and Taskar, B. (2002). Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707.

Goldenberg, A., Fienberg, S. E., Zheng, A. X., and Airoldi, E. M. (2010). A survey of statistical network models.

Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. (2006). Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*.

Jaeger, M. (2002). Relational Bayesian networks: a survey. *Linkoping Electronic Articles in Computer and Information Science*, 6.

Klinov, P. (2008). Pronto: A non-monotonic probabilistic description logic reasoner. In *The Semantic Web: Research and Applications*, pages 822–826. Springer.

Kunegis, J. and Lommatzsch, A. (2009). Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM.

Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

Lü, L. and Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170.

Lukasiewicz, T. and Straccia, U. (2008). Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):291–308.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Ochoa-Luna, J. E., Revoredo, K., and Cozman, F. G. (2011). Learning probabilistic description logics: A framework and algorithms. In Batyrshin, I. and Sidorov, G., editors, *Advances in Artificial Intelligence*, volume 7094 of *Lecture Notes in Computer Science*, pages 28–39. Springer.

Ochoa-Luna, J. E., Revoredo, K., and Cozman, F. G. (2013). Link prediction using a probabilistic description logic. *Journal of the Brazilian Computer Society*, pages 1–13.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference*. Morgan Kaufman.

Revoredo, K., Ochoa-Luna, J., and Cozman, F. (2010). Learning terminologies in probabilistic description logics. In da Rocha Costa, A., Vicari, R., and Tonidandel, F., editors, *Advances in Artificial Intelligence SBIA 2010*, volume 6404 of *Lecture Notes in Computer Science*, pages 41–50. Springer / Heidelberg, Berlin.

Sachan, M. and Ichise, R. (2011). Using semantic information to improve link prediction results in network datasets. *International Journal of COmputer Theory and Engeneering*, 3:71–76.

Schmidt-Schauß, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial intelligence*, 48(1):1–26.

Taskar, B., Wong, M.-F., Abbeel, P., and Koller, D. (2003). Link prediction in relational data. In *Proceedings of the 17th Neural Information Processing Systems (NIPS)*.

Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., and Zhang, X.-N. (2011). Link prediction for annotation graphs using graph summarization. In *The Semantic Web–ISWC 2011*, pages 714–729. Springer.

Wang, C., Satuluri, V., and Parthasarathy, S. (2007). Local probabilistic models for link prediction. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 322–331, Washington, DC, USA. IEEE Computer Society.

Wohlfarth, T. and Ichise, R. (2008). Semantic and event-based approach for link prediction. In *Proceedings of the 7th International Conference on Practical Aspects of Knowledge Management*.

Ontocloud – a Clinical Information Ontology Based Data Integration System

Diogo F.C. Patrão[1], Helena Brentani[2], Marcelo Finger[3], Renata Wassermann[3]

[1] A.C. Camargo Cancer Center

[2]Psychiatry Dept, Univ. of São Paulo

[3]Computer Science Dept, Univ. of São Paulo

djogo@cipe.accamargo.org.br, helena.brentani@gmail.com,

{mfinger,renata}@ime.usp.br

Abstract. *Relevant biomedical research relies on finding enough subjects matching inclusion criteria. Researchers struggle to find eligible patients due to: information scattered in many different databases, incompatible data representation, and the technical knowledge required to work directly with databases. We identified the required features of a clinical data search system and used it to design and evaluate Ontocloud, a prototype based on open source software and open standards of a dynamic ontology based database integration system with inference capabilities. A comparison between Ontocloud and three other database integration system showed that our prototype fulfilled its purpose and can be improved to be used in production.*

1. Introduction

The technology to quickly retrieve patient information from the Electronic Health Record is crucial to biomedical research. Traditional term based search techniques have been failing to bring accurate and precise results, due to the high complexity of this knowledge domain[Chard et al. 2011]. Database integration [Lenzerini 2002][Halevy 2001][Haas et al. 2002] provides techniques to consolidate information on several source databases through a set of mappings, into a single global database, which is then queried by the user. The most established database integration tools are based on relational databases, which are not tailored to deal with different conceptualizations of the source databases[Sujansky 2002].

Data collection for cancer research in a large hospital such as A.C. Camargo Cancer Center is hindered by a series of factors, the most important being: (1) Data is stored in many different databases in diverse ways, constantly changing and evolving; (2) Data is represented in a computer friendly format, hard to understand by physicians and scientists; (3) Collecting data manually is a time-consuming task, and clinical research projects need speed and accuracy on the recruit phase, (4) the same information may be present in different levels of detail, and (5) certain information is not explicitly asserted, but may be inferred based on indirect data.

In this work, we designed, implemented and evaluated a prototype of a database integration system called Ontocloud, based on open source software and standards. It addresses the issues (1)-(5), by providing some key features: dynamic access to data on

source databases; ontologies as the medium for data integration; and inference of concepts, harmonizing the detail level of similar information (the semantic mismatch issue), independence of source databases and data annotation. We describe how we implemented Ontocloud to solve a use case of integrating medical document metadata, and compare its characteristics against three other database integration architectures.

2. Background

2.1. Database integration

A database integration problem is described as taking several sources of complementary data and providing a single view for those sources[Halevy 2001]. In a theoretical perspective[Lenzerini 2002], we can represent a data integration system $\mathcal{I}$ as a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{G}$ is the global view, $\mathcal{S}$ is the set of source databases and $\mathcal{M}$ is the set of mapping functions from $\mathcal{S}$ to $\mathcal{G}$.

There are two methods for providing the global view $\mathcal{G}$: *dynamic* or *static*. Dynamic methods translate a query on global view $\mathcal{G}$ to queries on the relevant source databases $\mathcal{S}$ and translate back the answers using the mappings $\mathcal{M}$. Static methods (or data warehouse methods) create a materialized global view, by translating and copying all data from the sources $\mathcal{S}$ into a new database $\mathcal{G}$.

Both methods have their benefits and drawbacks. Dynamic methods rely on *query rewriting* or *query answering*, which are hard computational problems and therefore imply on slower performance. As they directly query the source databases, results are always up to date. Static methods are easier to set up and faster to query, however there is the need to translate all data on the sources and construct a new database before any queries can be answered. This procedure may require a higher level of access on the source databases and may take a great deal of time and disk space. Also, results are mostly always outdated, and the global database needs to be refreshed periodically[Halevy 2001].

Regarding the mappings, database integration systems can be classified as global as view (GAV) or local as view (LAV). Mappings on GAV systems transforms the source database into the global view, and queries are answered by several different algorithms[Halevy 2001]. Mappings on LAV systems maps the global view into the source, and in order to answer a query presented to the global view $\mathcal{G}$ the system should apply query answering (to infer results on $\mathcal{G}$ based on results on $\mathcal{S}$) or query rewriting (which translates the mappings from LAV to GAV). GAV mappings are easier for a developer to create than LAV mappings, however the former requires that all source databases are joined in one statement, being thus harder to add and remove sources than LAV. The query answering or rewriting step in a LAV system, depending on the complexity of mappgins, may demand a great deal of computation to be solved, if solvable at all; GAV systems relies on faster algorithms.

2.2. Ontologies, inference and database integration

An ontology represents knowledge in a formal framework, as concepts and relationships between pairs of concepts. Ontologies have been considered in heterogeneous database integration due to their ability to perform inferences and potential to deal correctly with the semantic mismatch problem [Wache et al. 2001] [Cruz and Xiao 2005].

Semantic mismatch is a problem that is intrinsic to data integration that usually leads to *loss of specificity* [Sujansky 2002] [Hull 1997]. It occurs when two sources of information have fields with similar but incompatible meanings. Usually, when it is necessary to join the two sources, the lowest level of detail should be adopted. In some cases of concept overlap it can be impossible to join sources. Ontologies, in data integration, mitigate information loss for some types of semantic mismatch. Figure 1 presents an example of such mismatch for information on patient smoking.

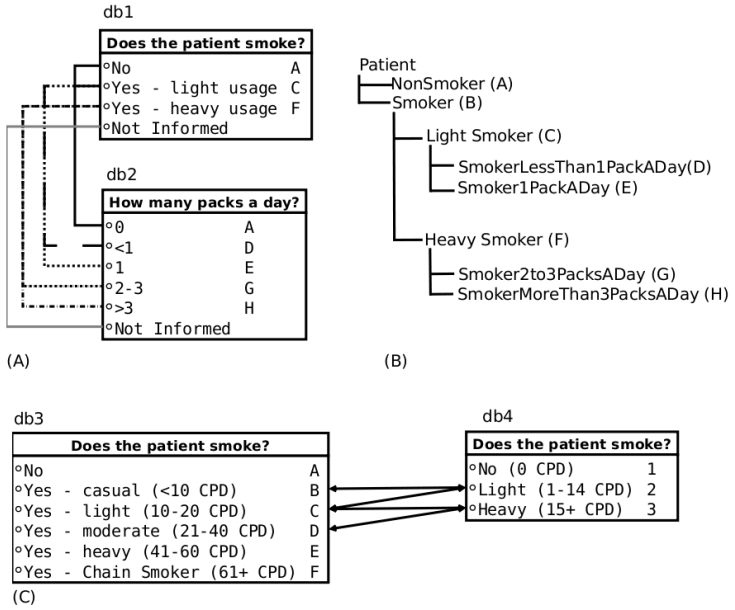


Figure 1. Semantic mismatch. (A) An example of a similar field in two different database $db1$ and $db2$; when the $db2$ field has value `0`, it is equivalent to a value of `N` on $db1$, `<1` and `1` is equivalent to `Yes - light usage` and `2-3` and `>3` to `Yes - heavy usage`. A reverse mapping would not be possible without loss of information specificity, because the options on db1 regarding light and heavy smokers might mean more than an option on db2. (B) An ontology that classifies all concepts involved on the source databases db1 and db2 from (A). Specific concepts such as $Smoker2to3PacksADay$ are classified under more general concepts, in this case, $HeavySmoker$ and $Smoker$. Instances of a specific class are considered also as belonging to its parent classes. (C) db3 and db4 contains an example of a semantic mismatch that is impossible to solve: note how the concept `Yes - light (10-20)` on $db3$ can be mapped to both `light (1-14 CPD)` and `heavy (15+ CPD)` on $db4$, at the same time that those two concepts on $db4$ maps each to two concepts on $db3$ (CPD - Cigarettes per day).

Ontologies can be represented in RDF/XML[1] format or in triplestores, which can be thought of as an equivalent of a database for ontologies. SPARQL[2] is the query language defined for querying data in an ontology. The SPARQL 1.1 specification allows for joining remote endpoints and thus integrating different datasets.

Inference is the process by means of which new information is derived from existing data from an ontology. Given abstract concepts, general rules can be added to a

[1]http://www.w3.org/TR/PR-rdf-syntax/
[2]http://www.w3.org/TR/rdf-sparql-query/

knowledge base to allow for new facts to be inferred[Russell and Norvig 2003]. An inference rule is divided in two parts, the head and the body. If the statements on the body is true, then the head statement will also be true. See Figure 2 for an example of an inference rule.

Query expansion [Bhogal et al. 2007] achieves inference by applying the rules over the query statements, instead of the facts of the knowledge base. A query $q_{\mathcal{G}}$ that specifies concepts presents on the head part of some inference rule may have this statement substituted by the body part of the rule (Figure 2).

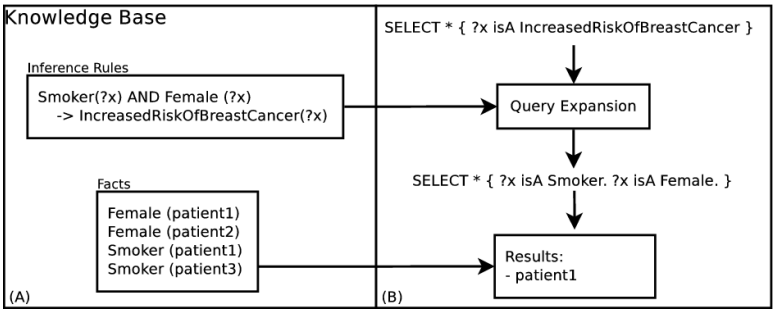


Figure 2. Inference by means of query expansion. (A) The inference rule states that if there is a patient $?x$ which belongs to class `Smoker` and `Female` (rule body), then this patient also belongs to class `IncreasedRiskOfBreastCancer` (head). (B) The term specified on the query is not stated as a fact on the Knowledge base, however a inference rule allows the terms to be substituted and the query can be answered.

3. Related work

Calvanese [Calvanese et al. 2007] describes Mastro-I, a data integration management system designed in order to maintain data complexity within reasonable bounds. It relies on the IBM product Infosphere Federation Server[3] to access source databases. In other work[Calvanese et al. 2011], the same group describes a database integration case using Mastro-I, in which five different data models were used, including XML-based and relational databases. The integration was made in two steps: first the different data models were combined using the InfoSphere Federation Server; then the Mastro-I system was used to map those entities into concepts, thus achieving data integration. In this architecture, there are two layers of heterogeneity solving: first, all relational data is mapped at the Federation Server, and then mapped into DL concepts, where integration is actually achieved.

DBOM [Cure and Bensaid 2008] is a GAV data integration system that uses decidable fragments of OWL language, OWL-DL and OWL-DL lite, to map results from queries over a relational database to an ontology. Several different relational sources can be used at once. It is able to deal with different degrees of confidence on each source, by configuring parameters on the mappings. It is implemented as a Protégé[4] plug-in, however, it is not cited whether this plugin is available, nor it has been found on the internet for download. As a use case the author presents the integration of two drug databases.

[3]http://www-01.ibm.com/software/data/infosphere/federation-server/

[4]http://protege.stanford.edu

The Query Integrator System (QIS) [Iller and Adkarni 2004] is a layer-based architecture that uses ontologies to represent and annotate metadata about the source databases; each change detected on the schemas generates annotations that can be reviewed later. It focuses on a dynamic environment where the source database schemas are constantly changing. Queries are composed by means of a visual tool that presents the annotation about the source databases, and translates these queries into SQL in the source databases.

Min et al [Min et al. 2009] integrated two sources of prostate cancer clinical data: one mantained by the Radiation Oncology department and the other from the Tumor Registry. The first contained data about radiotherapy treatment and the other demographic data. Both databases were integrated into one ontology by using a single D2R-Server instance. Integration was done by mapping concepts to two different databases in one single server. The integration was horizontal, as each database contained complementary data about one patient, except for one field, the TNM status, which was present in both.

Analyzing the available tools, none of them has features allowing to solve all of the clinical database integration issues we verified, except for Mastro-I and DBOM. However, the first relies on non-free software and it requires that relational sources are integrated first on a relational layer (the Infosphere Federation Server), and then on the ontology layer (Mastro). DBOM seems to be an interesting take on the subject, however it is not available anywhere for download. QIS has very interesting features but is based on obsolete standards and software.

4. Ontocloud design

Ontocloud was designed to provide dynamic access to a consolidated database global view of several database sources, using ontologies to consolidate heterogeneous data. Given a set of source databases $\mathcal{S}_{1..n}$, a set of source endpoint $\mathcal{E}_{1..n}$ should be provided. Each source publishes its objects of interest concepts of the global view $\mathcal{G}$ by means of a SPARQL endpoint. In order to get answers to a query $q_{\mathcal{G}}$ over the global database $\mathcal{G}$, the query must go through two transformation steps: the query expansion step accounts for inference, substituting terms not directly defined on the source endpoints; then the query federator step provides the query with SERVICE clauses that indicate in which source endpoint each concept is to be found (Figure 3).

Ontocloud uses four ontologies. The global ontology lists the classes and properties in which the global database will be represented, as well as annotations. The federation ontology specifies the source databases and which classes and properties of the global ontology they implement. The mapping ontology relates tables and columns from a source database to basic concepts on the global ontology. The inference ontology maps derived concepts to basic concepts through an ontology alignment file.

The global ontology should be the starting point when designing an ontology based database integration system, as the queries to be issued will refer to this ontology. It should be well anotated and descriptive, and should comprise the high-level concepts that will be queried as well as the ones actually on the source databases. Those are called *base concepts*, because they are directly related to a database object. The others are called *derived concepts* and should be related to base concepts by rules on the inference ontology.

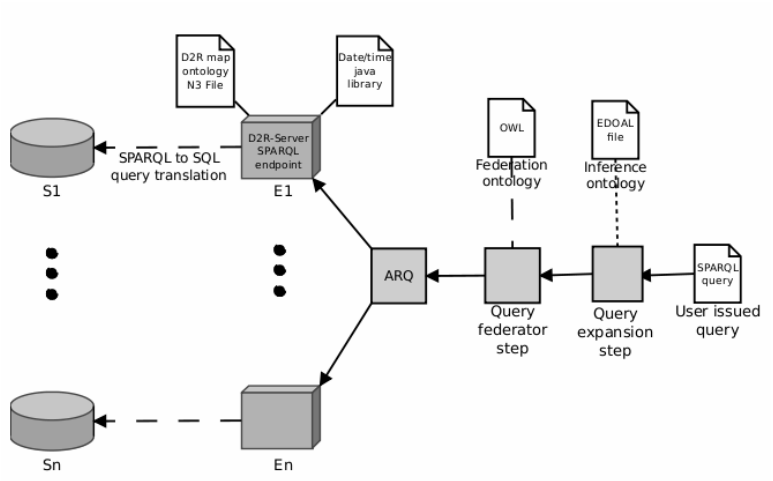


Figure 3. Ontocloud system architecture.

Each relational source database is required to have its own mapping file, which will translate access to the partial RDF graph of the global database $\mathcal{G}$ to SQL queries on the actual database objects.

The federation ontology lists all source databases and which concepts from the global ontology they provide. It is used by the federator step to translate a query $q_{\mathcal{G}}$ to a query $q_{\mathcal{E}}$ over the source endpoints.

Those characteristics make Ontocloud an adequate solution to integrating clinical databases, as stated in the introduction: (1) Integrated sources are independent, so adding, modifying or removing sources does not interfere with other sources; (2) The usage of ontologies allows for annotation of concepts, making it easier for a non-technically trained user to understand it; (3) Data is accessed directly from the sources, yielding always up-to-date results; (4) Ontologies provides tools for dealing with semantic mismatch; and (5) Inference of higher level concepts based on raw data, making all assumptions about data explicit and easy to audit.

4.1. Implementation

Ontocloud implementation was based on open standards and open source software. Its implementation is described in this section (illustrated in Figure 3). To map source databases as a SPARQL endpoint, we used D2R-server[Bizer and Seaborne 2004] with custom mapping N3 files. The query execution engine was ARQ, and custom software was implemented to perform the query expansion (to accomplish inference) and query federator (to indicate what are the databases to be looked into) steps.

D2RQ [Bizer and Seaborne 2004] is an OBDA[5] open source software. It is a Jena library that translates access to an RDF ontology specification by means of SQL queries, according to a mapping file. It includes D2R-Server, a server that provides a SPARQL endpoint over the mapped database, and dump-rdf[6], that converts the entire mapped database to a RDF file. Jena is a "Java framework for semantic web applica-

[5]Ontology Based Data Access
[6]http://d2rq.org/dump-rdf

tions"[7], providing an API for handling RDF, OWL, inference, triple storage and a query engine. JDBC[8] is a Java library that provides an unified API to access several different databases. D2R-Server did not provide any function for date and time operations, so we wrote custom Java classes and used it in SPARQL queries.

The Query Expansion Step used the inference ontology to translate queries using derived concepts into base concepts. We used the Mediation[9] library, which translates queries on an ontology A to an ontology B by means of an EDOAL [David et al. 2011] ontology alignment file. However, instead of mapping between two different ontologies, we mapped between concepts of the same ontology, avoiding circular references.

The Query Federator Step used the federation ontology to translate a query over the global ontology to the source endpoints. For each triple specified in the SPARQL query, it checks in which sources the concepts involved are present, and surrounds the triple with a SERVICE clause. If a concept is present in more than one source, it replaces the triple with a UNION of all SERVICE clauses. The software was written in Java using Jena library.

4.2. Use case

We selected as use case the problem of integrating clinical documents metadata from four information systems used at A.C. Camargo Cancer Center: **EHR**, which contains most data from clinic services; **Pathology**, that contains reports from anatomic pathology tests (visual inspection of sample tissues); **Image**, that contains reports from imaging tests; and **Prescriptions**, that contains both inpatient evolution (texts describing the patient's day-to-day evolution) and prescriptions of drugs and procedures.

We retrospectively consulted the Medical Informatics Laboratory ticket system, in which all query request made by doctors, managers and researchers are registered. Based on it, we compiled 17 queries of varying complexity to benchmark our integration system [10]. The Ethics Committee of A. C. Camargo Cancer Center, where this research was conducted, granted a waiver on informed consent. To answer those queries, we designed the global schema layout as depicted on Figure 4 and created the mappings accordingly.

We looked into the source databases for tables and columns that contained the needed information required by the defined global schema. Most databases contained all fields needed for the desired integration, except for the type of document on Pathology, Image and Prescription databases and the brazilian person registry number (CPF) for physicians on the Prescription database. We inquired physicians and discovered that documents on Pathology and Image databases are always reports and the documents on Prescription database can be a evolution or a prescription, depending whether a field is blank or not; the CPF number could be found for physicians which were linked to another database table, but not all of them (in this case, we simply created a new record without the CPF number).

To account for missing data, we created simple rules of inference based on knowledge provided by physicians. For Pathology and Image, all documents were stated to have

[7]http://jena.apache.org/

[8]http://www.oracle.com/technetwork/java/overview-141217.html

[9]https://github.com/correndo/mediation

[10]The queries are available at http://diogopatrao.com/ob/ as Supplementary Table 1.

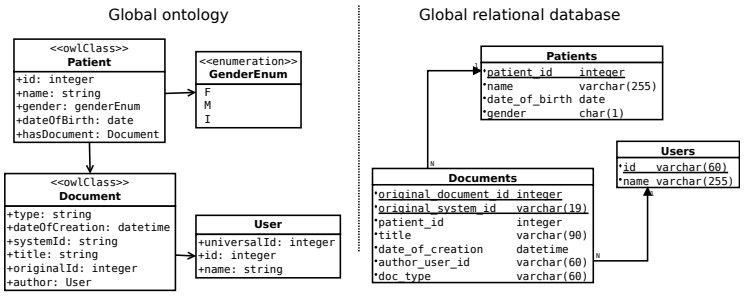


Figure 4. Global schema of our use case, for ontology and relational database integration architectures.

"PATHOLOGY REPORT" and "IMAGING EXAM REPORT" type. For Prescription, a conditional rule (based on whether a text field has data or not) was used to determine if a document belonged to "PRESCRIPTION" or "INPATIENT EVOLUTION" type. These rules were embedded on the mapping files.

It was possible to infer patient class based on the presence of certain types of documents on the patient's EHR; we implemented inference rules on the query expansion step using EDOAL ontology alignment file format. Examples of those rules can be found on Suplementary Table 2[11].

We replicated the original databases, by retrieving pertinent tables and columns and storing them into a single MySQL server. To extract the original sources into the MySQL database we used Pentaho Data Integration Community Edition[Golfarelli 2009]. It is a software suite to design and perform ETL (Extract, Transform, Load - a static database integration method). It allows one to graphically design scripts to extract data from several types of database, transform, mix and store them in a different database table or file.

5. Experimental setup

In order to assess performance and accuracy of Ontocloud we have set up three other database integration systems, which exhausts all combinations of the main database integration architecture characteristics: dynamic or static data acess, and relational database or ontology data representation. We evaluated accuracy in a qualitative way, by making sure that all 17 queries yielded equivalent results on all database integration systems evaluated. Query performance was evaluated as the total clock time a query took for completion on a integration system.

5.1. Source to global mapping

After replicating the source databases, we proceeded to set up all four database integration systems. The Supplementary Figure 1 depicts the experimental setup, and Supplementary Table 3 the database size and extraction times.

The tools used to set up the other integration systems are as follows:

[11] http://diogopatrao.com/ob/

- **Triplestore:** Openlink Virtuoso Universal Server Open Source Edition provides, among many other things, an RDF triple store and a SPARQL endpoint. We chose Virtuoso to implement Triplestore, the static access ontology based integration method. For each of the four source databases, we used D2R to dump data into an N3 file. Those files were imported using Virtuoso Bulk Loader[12].
- **Federation:** Teiid[13] is an open-source, dynamic relational database integrator system; it allows the creation of views over database resources published on a JBoss[14] server, and it is accessible as a JDBC resource. Federation, the dynamic database integration architecture, was designed as a Teiid instance. For each table in the global schema, we wrote a consolidated view, composed of queries over each source database joined by UNION clauses. Those queries did all necessary mapping to provide the required information, even if it was spread in different tables on the source database. The missing document type of Pathology, Image and Prescription databases was inferred directly on the view statement as a SQL constant or expression.
- **Replication:** The Replication architecture was created by materializing the Federation queries (translated to MySQL dialect) into tables. As in the source databases, every column in each database was indexed.

5.2. Experiments

The 17 queries were transcribed to each integration system language (SPARQL for ontology based systems and SQL for the others) and dialect (function names and namespaces were slightly different between MySQL and Teiid, and between Virtuoso and ARQ). There is no SERVICE specific optimizations on Jena, and we have not implemented it for Ontocloud. In contrast, Teiid, the software we chose for implementing Federation, was highly optimized for this type of queries. To account for this difference, we implemented two sets of queries for Ontocloud: one using both query expansion and query federator step, and other querying directly the sources with queries tuned by hand. This way, we get the actual running time for current software and an estimation of what the timing would be if there was an optimization step. We ran one single round of all 17 queries in all systems, without time limit and saving the results. To avoid server resource competition, only one one query on a single integration system was executed at a given time.

The computer server in which the experimental setup was created and tests were performed had 4 cores with 3.00GHz, 64bits, and 8GB of RAM, running CentOS 5. The database software installed was MySQL server version 5.0.95. We also used Pentaho Data Integration Community Edition version 4.0.1, ARQ-2.8.8, PHP 5.2.5, Virtuoso Open Source Edition 6.1.4.3127, D2R-Server 0.8, Java 1.6.0.23, Teiid 7.7 and JBoss 5.1.0 GA.

6. Results

A functional comparison between all systems can be seen on Table 1. All four integration systems were successfully configured and deployed. Except for queries 14 and 17 on Ontocloud Optimized, and queries 10-17 on Ontocloud Unoptimized, which were not

[12]http://www.openlinksw.com/dataspace/dav/wiki/Main/VirtBulkRDFLoader
[13]http://www.jboss.org/teiid/
[14]http://www.jboss.org/

completed due to lack of memory, all other queries on all evaluated systems completed successfully and yielded the same results. Ontocloud Optimized performed better than Federation on 7 queries out of 17, and was 15% faster than Ontoclound Raw (without optimizations). Replication was the fastest method of all, followed by Triplestore which performed better than Federation and Ontocloud on 13 queries. Time measurements for all database integration systems can be seen on Figure 5 and Supplementary Table 4.

Integration system	Data access strategy	Data heterogeneity solving method	Missing data	Annotation	Query expansion
Ontocloud	Dynamic	By ontology	Mapping	Yes	Yes
Federation	Dynamic	Least detailed	Mapping	No	No
Triplestore	Static	By ontology	Materialized	Yes	No
Replication	Static	Least detailed	Materialized	No	No

Table 1. Data integration architecture features.

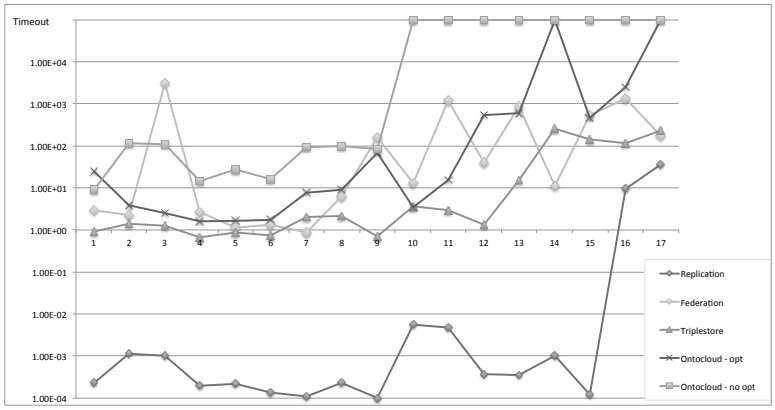


Figure 5. Time that each method took for running the 17 queries. The vertical axis unit is log_{10} seconds, and in the horizontal axis we display the query number. Query failures were plotted on the "Timeout" line, above all other measures.

7. Discussion

The implementation of Ontocloud and the use case experiments showed that it is an adequate database integration system for clinical data, as it accomplish the five objectives: (1) The configuration of source databases was completely independent, except for the Federation Ontology, which lists the URL of each endpoint and the concepts each implements; (2) The global ontology contained human-readable descriptions, so data would be easily understandable by non-technical personnel; (3) Data is accessed directly from the sources, yielding always up-to-date results; (4) Mappings provideded missing data in a way that is transparent to the end user and (5) Higher level concepts like TratedPatient and InPatient are easily understood by physicians and managers, while being translated by the query expansion step to its definition on raw data, allowing the query to be performed.

As we set up the integration systems, fundamental differences between Federation and Ontocloud arised. Federation requires that the developer explicitly join all sources in a single database view. That makes adding a new source to it a difficult and risky task, as it is required to work on a SQL statement that involves several different source databases

and any mistake may compromise the whole integration system. Each Ontocloud source is configured without the need to take other sources in consideration. Instead, it relies on the Query Federator step, which adds to the original query clauses indicating in which source endpoint each triple will be resolved. Therefore, by keeping the mapping files separated, Ontocloud facilitates the maintenance of source databases.

Inference on Ontocloud was based on the Mediation library, which allowed us to implement rules by expanding each query term. The inference rules are detached from the database integration itself, and can be maintened independently of the sources. Also, as those rules are represented on an ontology language, it is more suitable for domain experts to maintain it than on the relational methods, in which rules should be implemented on SQL language. It also improves the information management of such a system, as it keeps the raw data (on the mapping ontologies) apart from the higher level concepts (on the inference ontology).

Ontocloud performance suffered on queries with aggregation or that dealt with date operations. This occurs because SPARQL aggregation keywords and date manipulation functions are not translated directly to SQL, instead all results are retrieved and transformations are performed in memory. That both hindered performance and required a lot of memory. Also the queries generated by Query Federator step contained a lot of SERVICE keywords, each containing only one triple. An important optimization would be to join triples on the same SERVICE pattern, minimizing the access to source endpoints. Also, the order of triples and filters on the SPARQL query is crucial to determine the performance. Those optimizations are beyond the scope of this work, but would certainly put Ontocloud on a par with the other methods. For the purpose stated in this work, the speed of Ontocloud seems a fair tradeoff for the ability of yielding up-to-date results at any time and performing inference.

8. Conclusion

We have successfully designed and implemented Ontocloud to perform ontology-based database integration. It implements important features in an clinical data integration system: The sources are loosely coupled, favoring distributed and dynamic management of sources; uses ontologies to integrate data, which is prone to reuse and more human readable; has dynamic access to sources, always yielding up-to-date results; and allows inference. We believe that this system architecture can be extended and improved, as indicated in the discussion, to become a production level tool very useful in the medical informatics context.

References

Bhogal, J., Macfarlane, A., and Smith, P. (2007). A review of ontology based query expansion. *Information Processing & Management*, 43(4):866–886.

Bizer, C. and Seaborne, A. (2004). D2RQ-treating non-RDF databases as virtual RDF graphs. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., and Savo, D. F. (2011). The MASTRO system for ontology-based data access. *Semantic Web*, 2(1):43–53.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., and Rosati, R. (2007). Mastro-i: Efficient integration of relational data through dl ontologies. In *Proc. of the 20th Int. Workshop on Description Logics (DL 2007)*, volume 250 of *CEUR Electronic Workshop Proceedings, http://ceur-ws.org/*, pages 227–234.

Chard, K., Russell, M., Lussier, Y. A., Mendonça, E. A., and Silverstein, J. C. (2011). A cloud-based approach to medical NLP. *AMIA - Annual Symposium proceedings / AMIA Symposium*, 2011:207–16.

Cruz, I. F. and Xiao, H. (2005). The role of ontologies in data integration. *Journal of engineering intelligent systems*, 13(4):854–863.

Cure, O. and Bensaid, J.-D. (2008). Integration of relational databases into OWL knowledge bases: demonstration of the DBOM system. In *2008 IEEE 24th International Conference on Data Engineering Workshop*, pages 230–233. IEEE.

David, J., Euzenat, J., Scharffe, F., and dos Santos, C. T. (2011). The alignment API 4.0. *Semantic web*.

Golfarelli, M. (2009). Open Source BI Platforms: A Functional and Architectural Comparison. In Pedersen, T., Mohania, M., and Tjoa, A., editors, *Data Warehousing and Knowledge Discovery*, volume 5691 of *Lecture Notes in Computer Science*, pages 287–297. Springer-Verlag.

Haas, L. M., Lin, E. T., and Roth, M. A. (2002). Data integration through database federation. *IBM Systems Journal*, 41(4):578–596.

Halevy, A. Y. (2001). Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294.

Hull, R. (1997). Managing semantic heterogeneity in databases. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS '97*, pages 51–61, New York, New York, USA. ACM Press.

Iller, P. E. L. M. and Adkarni, P. R. N. (2004). QIS : A Framework for Biomedical Database Federation. *Journal of the American Medical Informatics Association*, 11(6):523–534.

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, page 246. ACM.

Min, H., Manion, F. J., Goralczyk, E., Wong, Y.-N., Ross, E., and Beck, J. R. (2009). Integration of prostate cancer clinical data using an ontology. *Journal of biomedical informatics*.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, 3rd edition.

Sujansky, W. (2002). Heterogeneous Database Integration in Biomedicine. *Journal of Biomedical Informatics*, 34(2001):285–298.

Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S. (2001). Ontology-Based Integration of Information A Survey of Existing Approaches. In *IJCAI-01 Workshop: Ontologies and Information Sharing*, volume 2001, pages 108–117.

An Automated Transformation from OntoUML to OWL and SWRL

Pedro Paulo F. Barcelos[1], Victor Amorim dos Santos[2], Freddy Brasileiro Silva[2], Maxwell E. Monteiro[3], Anilton Salles Garcia[1]

[1]Electrical Engineering and [2]Computer Science Departments
Federal University of Espírito Santo - UFES
Vitória – ES – Brazil

[3]Federal Institute of Espírito Santo - IFES
Serra – ES – Brazil

{pedropaulofb, victor.amsantos, freddybrasileiro}@gmail.com,
maxmonte@ifes.edu.br, anilton@inf.ufes.br

Abstract. *OntoUML and OWL are ontology languages appropriated to different knowledge representation levels. In order to have better knowledge representation and reasoning capabilities in OWL ontologies, an Ontology Engineering should be used – which corresponds to the transformation of a conceptual model ontology language, such as OntoUML, to a computational ontology language, such as OWL. This paper aims to bridge the expressivity gap between these languages through a Model Driven Architecture automated transformation from OntoUML to OWL with SWRL rules that contributes to (i) make easier the OWL creation from OntoUML, (ii) eliminate the human errors in this process, (iii) improve the resultant OWL ontology semantics.*

1. Introduction

In order to have better knowledge representation and reasoning capabilities in computational ontologies, like the ones represented with the Web Ontology Language (OWL), an Ontology Engineering with well-defined phases is defended in [Guizzardi 2007]. In a conceptual modeling phase, highly-expressive languages should be used to create strongly axiomatized ontologies that approximate as well as possible to the ideal ontology of the domain. The focus of these languages is on representation adequacy, since the resulting specifications are intended to be used by humans in tasks such as communication, domain analysis and problem-solving [Guizzardi 2007]. Guizzardi proposed in [Guizzardi 2005] an ontologically well-founded profile of the Unified Modeling Language (UML), later named OntoUML, to be a language used in this step. OntoUML provides stereotypes based on the Unified Foundational Ontology (UFO) to capture domain knowledge and has been successfully applied in different domains like electrophysiology [Gonçalves et al. 2007], telecommunications [Barcelos et al. 2011] and oil and gas [Guizzardi et al. 2010].

Once users have already agreed on a common conceptualization, versions of a reference ontology can be created as the objective of the Ontology Engineering (its last phase). These versions have been named in the literature lightweight ontologies. Contrary to reference ontologies, lightweight ontologies are not focused on

representation adequacy but are designed with the focus on guaranteeing desirable computational properties [Guizzardi 2007]. An Example of a language suitable for lightweight ontologies is the Web Ontology Language (OWL). OWL is the standard language for knowledge representation and reasoning in the semantic web and in computational applications. The addition of rules written in Semantic Web Rule Language (SWRL), a Horn-like rule language, in OWL ontology improves its representation expressivity.

In order to achieve this objective, an intermediate phase is necessary in the Ontology Engineering: a phase to bridge the gap between the conceptual modeling of references ontologies and the coding of these ontologies in terms of specific lightweight ontology languages. Issues that should be addressed in such a phase are, for instance, determining how to deal with the difference in expressivity of the languages that should be used in each of these phases [Guizzardi 2007]. This paper aims to present an automated transformation from an OntoUML model to OWL ontology with SWRL rules, here named OntoUML2OWL+SWRL, which is inserted into this Ontology Engineering phase.

The OntoUML2OWL+SWRL is a Model Driven Architecture (MDA) transformation that contributes to the creation of OWL files with improved semantics to be used for knowledge representation and reasoning on computational applications. Two different OntoUML to OWL transformations already exists; however, OntoUML2OWL+SWRL differ from them in scope and complexity.

This paper is structured as follows: Section 2 presents the OntoUML2OWL+SWRL, including all conceptual considerations and limitations, and all the implementation technologies used. As related works, Section 3 presents the other OntoUML to OWL transformations and their relations to our transformation. Section 4 presents some conclusions as well as future works. Background information about OntoUML and OWL concepts is provided during the paper's sections.

2. The OntoUML2OWL+SWRL Transformation

The OntoUML2OWL+SWRL transformation was created as a Model Driven Architecture (MDA) transformation [Miller and Mukerji 2003]. This transformation is done in the M2 level (the metamodel level), which makes it reusable, as each specific transformation in the M1 level (the domain model level) is an instance of the generic M2 transformations. The conceptual ontology model can be seen as a Computational Independent Model (CIM), while the OWL with SWRL rules model can be seen as a Platform Independent Model (PIM). Further transformations can be created from the PIM (the OWL) to code - a possible Platform Specific Model (PSM). OntoUML metamodel is presented in [Guizzardi 2005], and a MOF-Based OWL metamodel can be found in http://www.w3.org/2007/OWL/wiki/MOF-Based_Metamodel.

OntoUML2OWL+SWRL accomplish the following objectives: (i) make easier the OWL files creation from OntoUML models, (ii) eliminate the human errors in this process, and (iii) improve the resultant OWL ontology semantics.

The conceptual transformation' considerations are presented in section 2.1, while the implementation tools and languages are presented in section 2.2.

2.1. The Conceptual Transformation's Design

An intrinsic characteristic of transformation from high expressive modeling languages to computational ones (that must be decidable, tractable, etc.) is the loss of expressivity. These losses are presented as limitations during this section. We can cite, as a first example, the incapacity of this transformation to represent OntoUML's *existential dependencies* (specific instance dependence). Although OWL can represent existential dependencies, in order to allow this representation, the classes' instances must be known. As no instances are represented in OntoUML models, the transformation cannot create the resulting OWL with the existential restrictions.

The design considerations about OntoUML2OWL+SWRL transformation are described in this section. Our intention here is to hide as much as possible the resulting code and present just the ideas.

Classes: We have taken as a development premise the separation of the models' concepts with the metamodel's ones for class transformation. That is, in OntoUML2OWL+SWRL the generated OWL file contains only domain classes, for example, applying the transformation to a Genealogy OntoUML model, the resulting OWL will have just classes with Genealogy concepts, like Mother, Father and Offspring. It will not have OntoUML metamodel's concepts like Kind, Role, etc. This decision simplifies the generated OWL and makes it simpler to the users (humans or machines).

In classes' transformation, the OntoUML classes are directly translated to OWL classes. Even though the simplicity of this transformation, OntoUML's metamodel restrictions are considered in this step. Disjoint concepts and *Phases-partitions* (a special kind of generalization sets), explained hereafter, are examples of these considerations.

Disjoint Concepts: One of UFO's meta-properties is the *identity principle*, which is related to the nature of an object. For example, a Student is a Person, as they have the same identity principle, but they can never be a Horse, as these entities have different identity principle. The entities that provide identity principles are named Substance Sortals (stereotyped in OntoUML as Kinds, Quantities or Collectives). Mixins (Categories, Role Mixins or Mixins, in OntoUML) are the entities that aggregate objects of different identity principles. An example of Mixin is the concept "Animal", as it aggregate instances of the classes Person and Horse. In contrast with Sortals and Mixins, Moments (Modes and Relators) are entities that inhere in, and, therefore, are existentially dependent of, another entity. These entities' restrictions are considered in the OntoUML's metamodel, as can be seen in Figure 1 below.

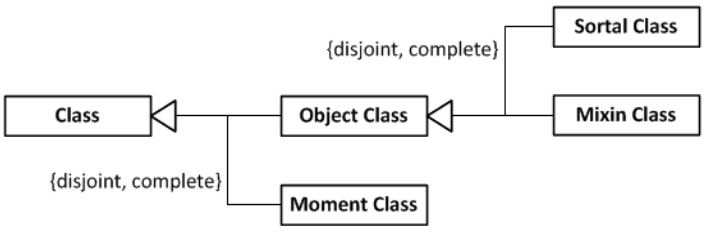


Figure 1 - Fragment of the OntoUML's metamodel

The disjoint entities are implemented in OntoUML2OWL+SWRL by the following considerations (top-level entities are entities that are not generalized by others): (a) all Substance Sortals are disjoint from each other; (b) all top-level Moments are disjoint from each other; (c) top-level Moments are disjoint from Substance Sortals and from top-level Mixin Class types.

Generalization Sets: OntoUML have two generalization sets' meta-properties: *isCovering* and *isDisjoint*, both of Boolean type. These meta-properties were considered in this transformation as follows:

- *isCovering* = *true*: the generalized class is equivalent to all complete set.
- *isDisjoint* = *true*: the generalizing classes are marked disjoint from each other.

Figure 2 presents as an example: (a) an OntoUML generalization set, (b) the resultant OWL class taxonomy, (c) the OWL Class' Person definition, and (d) the OWL Class Man's definition.

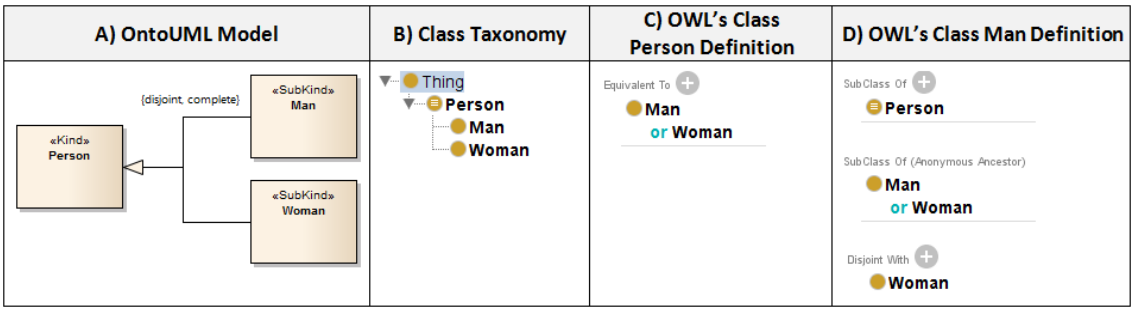


Figure 2 – Transformation of Generalization Sets

In OntoUML, *Phases-partitions* are a special type of generalization sets composed of classes stereotyped as Phases. As a particularity, they have always the true value for *isDisjoint* and *isCovering*. This particularity is considered in OntoUML2OWL+SWRL transformation.

Associations: OWL distinguishes between two main categories of associations, called *properties*: *Object properties*, that link individuals to individuals, and *DataType properties*, that link individuals to data values [Hitzler et al. 2012]. In OntoUML2OWL+SWRL, OntoUML associations are mapped to Object properties (here discussed), while DataTypes are mapped to Data properties (discussed later in this section).

Differently from OntoUML, which do not have directed associations, OWL properties are directed binary relations. This implies the necessity to create two object properties for each OntoUML association: a direct one and its inverse. As a design choice, we have named the inverse relation with the same direct relation's name prefixed with "INV.". This decision was taken because the generation of improved inverse names (for example: "drives" and "is driven by") would require language processing and it would be different in every natural language (English, French, etc.).

OntoUML associations always have a source class and a target class. Source and Target classes are considered in the transformation in order to create, respectively, the domain and range of an OWL object property. The nomenclature of generated OWL

object property is also related to these classes, as the reading direction is not a feature of OntoUML metamodel, i.e., it is just a visual resource and cannot be read from the OntoUML model to the OWL ontology. In order to produce the desired OWL object property name, the name of the OntoUML model must be given from the source class to the target one. Figure 3 illustrate the results of correct and incorrect associations.

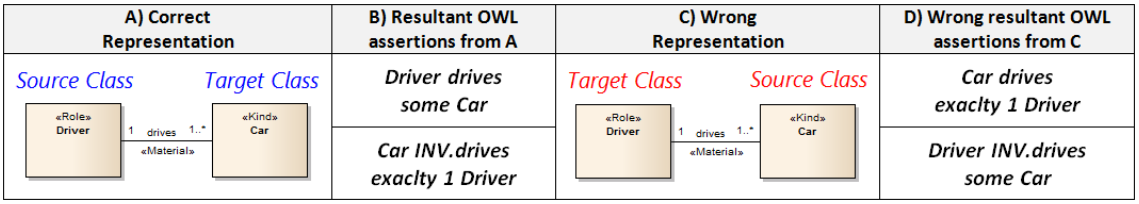


Figure 3 - Association representation

When no name is assigned to an association, the association's OntoUML stereotype is used to create its name using the following nomenclature: "AssociationStereotype.SourceClassName.TargetClassName". An example of a relation named this way can be found in the SWRL rule found in Figure 4.

Every object property is asserted as Equivalent Class of the class that it is related, except in the case when the cardinality's lower bound is zero (explained in *Cardinalities*). Disjointness of object properties is also considered as relations with different stereotypes are set as disjoint from each other (associations with the same stereotype are not set as disjoint from each other, as one can be a specialization of other). OntoUML's Material and Part-whole relations are separately explained as their transformations have particularities.

Material Relations: In OntoUML, Material relations are the ones that depend on a Relator to exists, i.e., the Material relations are derived relations that need a truth maker to exist. Figure 4 (A) presents a Material relation ("drives") that is derived from the existence of the Relator License.

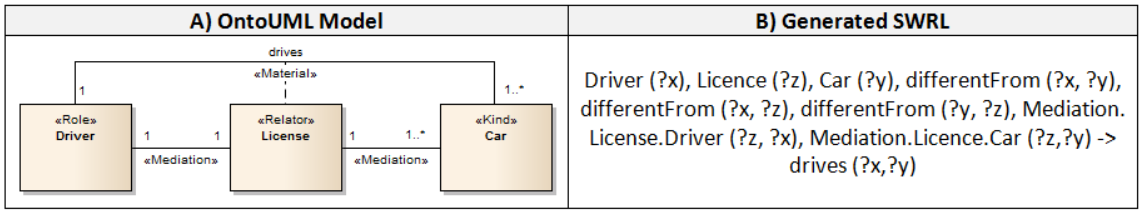


Figure 4 - SWRL resultant from Material relations

To each Material relation that exists in an OntoUML model a SWRL rule is created. This rule aims to represent the Material relation's derivation from the Relator.

Every SWRL rule created in this transformation is in accordance with Description Logic (DL) safe-rules [Motik et al. 2005], guaranteeing reasoning decidability.

Part-whole Relations: Differently to other associations, Part-whole relations are transformed to OWL sub-object properties of an object property with the name of its stereotype. This is done in order to better represent its meta-properties (called characteristics in OWL).

According to UFO, part-whole relations (stereotyped as *componentOf*, *memberOf*, *subCollectionOf* and *subQuantityOf* in OntoUML) are always irreflexive and asymmetric – a characteristic that is considered in OntoUML2OWL+SWRL.

Part-whole relations' different types have different transitivity relations, as can be seen in [Guizzardi 2005]. *subCollectionOf* and *subQuantityOf* are transitive; *memberOf* is intransitivity (it is never transitive); *componentOf* is non-transitivity, i.e., there are cases when it is transitive and other cases when it is not. Figure 5 represent the transitivity cases considered in OntoUML2OWL+SWRL (empty stereotypes are left to indicate that the pattern can occur with the following functional complex stereotypes: Kind, Subkind, Role or Phase).

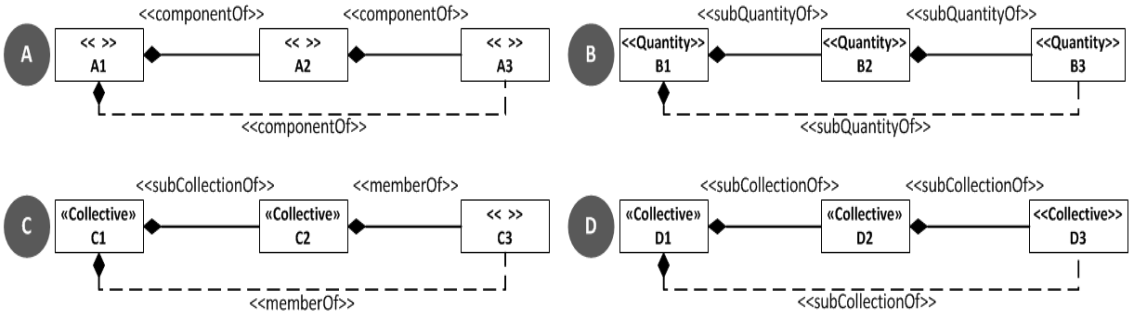


Figure 5 - Transitivity cases considered in OntoUML2OWL+SWRL

Four different generic SWRL rules can be created to represent the transitivity cases from Figure 5. These rules are added to the resultant OWL ontology when its specific case occurs. For example, every time the transitivity case (A) from Figure 5 can occur (the sum of *componentOf* is greater than 1), the following SWRL rule is created: *componentOf (?x, ?y), componentOf (?y, ?z), differentFrom (?x, ?y), differentFrom (?x, ?z), differentFrom (?y, ?z) -> componentOf (?x, ?z)*.

It is important to note that SWRL rules acts over instances, while the object properties' characteristics are defined in a higher level in OWL. If we just mark, for example, *subCollectionOf* as irreflexive, asymmetric and transitive, this will result in an error. As in the SWRL rules we are stating that the transitivity occurs only in different elements (by using the *differentFrom* operator), this error does not occurs.

An important limitation on OntoUML part-whole relations representation is about its metaproperties *isEssential* and *isInseparable*, which cannot be represented in OWL as they represent the existential dependence between parts and wholes.

DataTypes: Direct and structured DataTypes, with and without asserted cardinality, are treated in our transformation, as presented in Figure 6. These DataTypes are mapped to OWL's DataType properties.

Figure 6 - Example of considered different representations of DataTypes

The transformation supports the following OWL DataTypes: unsigned int, unsigned byte, double, String, normalized string, Boolean, hex binary, Integer (int), short, byte, unsigned long. If the provided DataType is not one of these, the transformation creates it as a Literal. Hidden cardinality is mapped to "exaclty one" concept in OWL. Attributes from the same class are set as disjoint from each other.

Applying the OntoUML2OWL+SWRL transformation to the model presented in Figure 6 we have the following object properties presented in Figure 7.

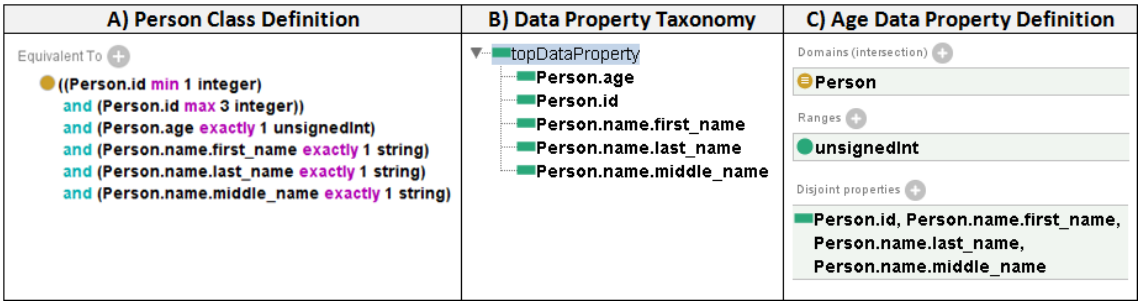


Figure 7 – OntoUML's DataTypes transformation to OWL Data Properties

DataTypes are created with the following nomenclature: "*Class.AttributeName*". In case of a structured DataType, it is created with the following nomenclature "*Class.AtributeName.StructuredDatatypeAtributeName*".

Cardinalities: Different cardinalities imply different transformations, as can be seen in Figure 8. This holds for object properties as well as to DataType properties.

OntoUML Cardinality	OWL Class Assertion	OWL Cardinality Restriction
0..*	Not treated	
0..N (0 < N < *)	Subclass Of	Max N
N (N ≠ 0)	Equivalent Class	Exactly N
1..*	Equivalent Class	Some
N..M (1 ≤ N < M, M ≠ *)	Equivalent Class	Min N and Max M

Figure 8 – Cardinality transformation

As can be seen in Figure 8, there's a transformation limitation to represent cardinalities with lower bound equal to zero, since the assertion "*has min 0*" would provoke an inconsistency. This happens because in OWL all elements "*have min 0*" properties with any other element, hence, OWL assumes that any instance of a class may have zero or more values for a particular property since a restriction was not added [Patel-Schneider et al. 2004].

In fact, properties (associations and attributes) with minimum cardinality 0 (optional properties) are not desirable in OntoUML models as they usually hide an entity's role. For example, an association "Person drives 0..* Car" hides the Person's role Driver. As stated in [Guizzardi 2005], the representation of optional cardinality constraints leads to unsound models with undesirable consequences in terms of clarity.

2.2. Transformation Implementation Technologies

The Ontology Lightweight Editor (OLED), currently in its version 0.8, is more than just an OntoUML editor - it is full framework for development of OntoUML ontologies. It provides: (a) a model editor , (b) a syntactical validation, (c) an OntoUML to OWL transformation, (d) a validation environment, which provides semantic validation realized as anti-pattern identification and treatment, and as a visual simulation through an Alloy transformation [Sales et al. 2012]. OLED is a free tool available for download at: https://code.google.com/p/ontouml-lightweight-editor/.

We have taken as a requisite to the development of the OntoUML2OWL+SWRL that the generated OWL file must open in Protégé 4.3. This decision was taken due to the fact that the Protégé is the most used tool for creation of OWL ontologies - it can be helpful to developers to view the OWL resultant from the transformation.

We have used as implementation language Java and, in order to do the translation, we have used The OWL API (http://owlapi.sourceforge.net/), an open source API that allows the developer to easily create OWL files.

As a usability issue, it is important to mention that, for OntoUML2OWL+SWRL, it is not obligatory that the OntoUML models to be created directly on OLED. The models can be created on professional tools like Sparx Systems Enterprise Architect or at Astah and exported as an XMI file and them imported in OLED.

OntoUML2OWL+SWRL code is open and can be found inside OLED's project.

3. Other OntoUML to OWL Transformations

The first identified initiative to create an OWL ontology with SWRL rules codification from an OntoUML model were made in [Zamborlini et al. 2008]. Although this transformation has been used to create an application based on an heart's electrophysiology ontology [Gonçalves et al. 2007], no automated transformation was created from the OntoUML model to the OWL, i.e., the OWL ontology was created manually.

Two other OntoUML to OWL transformations already exists (none of them considers SWRL rules), both implemented at the Ontology Lightweight Editor (OLED). In this section we are going to discuss the conceptual aspects of these two different transformations: the OLED's Simple Transformation (Section 3.1) and the Temporal Transformation (Section 3.2).

In order to exemplify the differences between the transformations, we are going to consider the following OntoUML model, presented in Figure 9. This simple OntoUML model does not intent to represent the world as it is: it is just a syntactical valid model with simple concepts in order to be used as a valid input for the transformations presented in this paper. This diagram states that every Person has Headache and that Persons can be Drivers. To be a Driver the Person has to be related with one License that is related with one or more Cars. The Protégé 4.3 software (http://protege.stanford.edu/) was used to visualize the generated OWL ontologies.

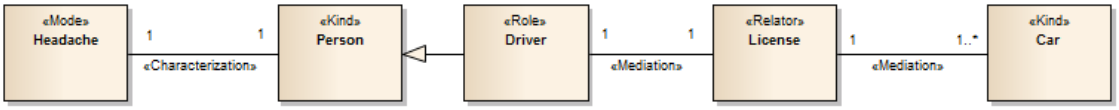


Figure 9 - Simple OntoUML model used as example

3.1. *OLED`s Simple Transformation*

The OLED's Simple Transformation, implemented by researcher Antognoni Albuquerque, was the first transformation from OntoUML to OWL that (similarly to the transformation proposed in this paper) did not included OntoUML stereotypes in the resultant OWL file. OLED's Simple Transformation treats the following cases in a different manner then we do: generalization set meta-properties and disjoint classes based on OntoUML stereotypes. It does not, however, treats: part-whole relations' meta-properties (nonetheless the user can create them using annotations in the OntoUML model) and transitivity, material relations' derivations, and structured DataTypes (it does treats simple DataTypes, creating OWL data properties). Another important design difference from this transformation to OntoUML2OWL+SWRL is the fact that it creates the OWL axioms as "subClassOf" instead of "EquivalentClasses". OLED's Simple Transformation does not consider temporal aspects.

Figure 10 represents, for the example model presented in Figure 9, the OLED's simple transformation for: (a) the class taxonomy, (b) the Object property taxonomy and (c) the License class description.

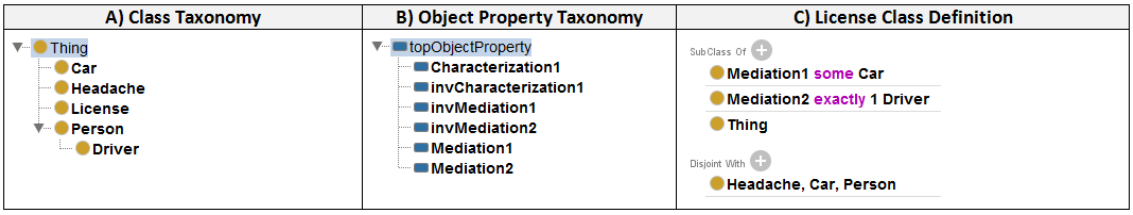


Figure 10 – OLED's Simple Transformation results

Considering the available transformations, the OLED's Simple Transformation is by far the most similar transformation to OntoUML2OWL+SWRL as both do not consider temporal aspects and as both do not intent to represent OntoUML or UFO (the foundational ontology which OntoUML is grounded) concepts in the generated OWL file. Yet, still comparing both transformations, OLED's Simple Transformation lacks in expressivity in comparison to OntoUML2OWL+SWRL as the latter considers more OntoUML restrictions when creating the OWL result.

3.2. *Temporal Transformation*

Similarly to this paper, [Zamborlini 2011] proposes alternatives for an OntoUML to OWL transformation concerned in to represent ontologies in an epistemological level language representing all ontological distinctions in order to guarantee the model quality. However, [Zamborlini 2011] has focus on temporal questions in the transformation process, while only static world is considered in the OntoUML2OWL+SWRL Transformation. Zamborlini's transformation is called here *Temporal Transformation* and it is also available in OLED.

The Temporal Transformation has two different approaches to consider temporal aspects of OntoUML in OWL. These approaches are (a) the Reification and (b) the Worm View.

a) Reification Transformation

Reification can be understood as the objectification of something so one can refer to it, qualify it and quantify it.

The focus in this transformation is the ontological difference between Objects and Moments, in which mutable information of individuals are reified. Then, the reification covers different types of moments. In this way, every others entities are mapped as Objects.

Applying the Reification Transformation to the OntoUML example model presented in Figure 9 we can see (a) the class taxonomy, (b) the Object property taxonomy and (c) the License class description in Figure 11.

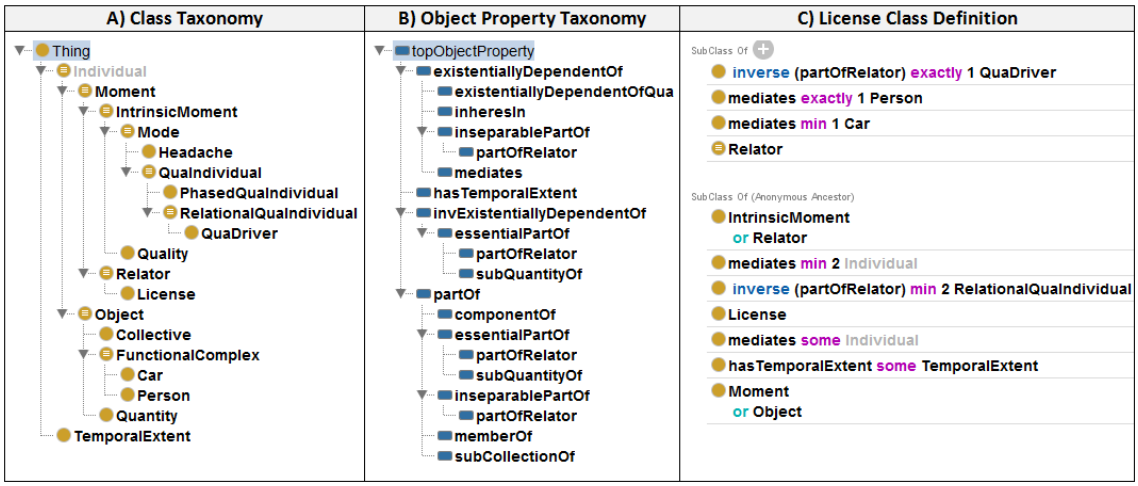


Figure 11 - Temporal Reification Transformation results

b) Worm Views Transformations

The OLED tool implements three different Worm View temporal considerations over OWL, they are called A0, A1 and A2.

In this approach individuals are considered spatiotemporal worms whose temporal parts are worm slices, in a way that individuals are composed by temporal parts and individual concept that maps him. The OWL base structure to represent this approach is divided in two different levels, the static one, called the *Individual Concept Level* (ICL), and the dynamic one, called the *Time Slice Level* (TSL). These three implementations are:

A0: Rigid concepts are represented in ICL; other concepts, relations, attributes in TSL.

A1: Rigid concepts, necessary and immutable attributes, and relations that implies in mutual existential dependency are represented on the ICL, and concepts, relations that not implies in mutual existential dependency and attributes not necessary and immutable simultaneously, on the TSL.

A2: Rigid concepts, necessary and immutable attributes, and relations that implies in existential dependency are represented on the ICL, and concepts, relations that not implies in existential dependency and attributes not necessary and immutable simultaneously, on the TSL.

As can be noticed, the Temporal Transformation has huge different considerations from OntoUML2OWL+SWRL because it is focused in the representation of temporal aspects. These transformations present a more expressive OWL as a result, but to do this it mix domain concepts with OntoUML and UFO concepts (see Figure 11) which demands that the OWL user (a person or a computational application) has this previous knowledge in order to understand and manipulate the output of the transformation. OntoUML2OWL+SWRL have as premise that just domain concepts are created in the OWL, resulting in a comprehensive OWL file.

4. Conclusions

This paper presents a Model Driven Architecture automated transformation from OntoUML to OWL with SWRL rules, named OntoUML2OWL+SWRL, that contributes to (i) make easier the OWL creation from OntoUML, (ii) eliminate the human errors in this process, (iii) improve the resultant OWL ontology semantics. This transformation is placed between two phases of an Ontology Engineering as it bridges the gap between two classes of languages with different purposes: (i) OntoUML, on one hand, is a well-founded ontology representation language focused on representation adequacy regardless of the consequent computational costs, which is not actually a problem since OntoUML models are targeted at human users; and (ii) on the other hand, OWL, a lightweight representation language with adequate computational properties.

Although two other OntoUML to OWL transformation exists (namely, the Simple OLED's transformation and the Temporal Transformation) OntoUML2OWL+SWRL have different transformation scope and it is placed between them in complexity. Differently from the other existent transformation, OntoUML2OWL+SWRL also create SWRL rules for representation of Mediations and Part-whole relations.

The conceptual transformation's design was presented with limitations and other implications inherent to these kinds of transformations. OntoUML2OWL+SWRL is implemented in OLED, a framework for OntoUML, and its code is open and fully available.

As the required OWL expressivity can be different depending on its application, the implementation of a parameterized transformation, where the user can choose which features the resulting OWL can have, is a future work. Also, transformation for specific OWL profiles can be created. As visual diagramming languages (including here OntoUML) are not always able to capture all relevant restrictions of a domain, they are usually incremented with restriction rules in Object Constraint Language (OCL). The coupling of an OCL to SWRL transformation to OntoUML2OWL+SWRL is desired. The DataType transformation can be improved in the future considering extensions to UFO presented in [Albuquerque and Guizzardi 2013], where the notion Semantic Reference Spaces to are employed to improve the ontological foundations concerning value spaces.

Acknowledgements. This research has been funded by FAPES/CNPq (PRONEX 52272362/11).

References

Albuquerque, A. and Guizzardi, G. (2013). An Ontological Foundation for Conceptual Modeling Datatypes based on Semantic Reference Spaces. In 7th IEEE International Conference on Research Challenges in Information Science (RCIS 2013).

Barcelos, P. P. F., Guizzardi, G., Garcia, A. S. and Monteiro, M. E. (may 2011). Ontological Evaluation of the ITU-T Recommendation G.805. In 2011 18th International Conference on Telecommunications. IEEE.

Gonçalves, B., Guizzardi, G. and Filho, J. G. P. (2007). An electrocardiogram (ECG) domain ontology. In 2nd Workshop on Ontologies and Metamodels for Software and Data Engineering.

Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. Enschede: Centre for Telematics and Information Technology University of Twente.

Guizzardi, G. (2007). On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. Proceedings of the 2007 conference on Databases and Information Systems IV: 7th International Baltic Conference, p. 18–39.

Guizzardi, G., Baião, F., Lopes, M. and Falbo, R. (2010). The Role of Foundational Ontologies for Domain Ontology Engineering: An Industrial Case Study in the Domain of Oil and Gas Exploration and Production. International Journal of Information System Modeling and Design, v. 1, n. 2, p. 1–22.

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F. and Rudolph, S. (2012). OWL 2 Web Ontology Language Primer (Second Edition). http://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

Miller, J. and Mukerji, J. (2003). MDA Guide Version 1.0.1. Object Management Group, http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf.

Motik, B., Sattler, U. and Studer, R. (2005). Query Answering for OWL-DL With Rules. Web Semantics: Science, Services and Agents on the World Wide Web 3.1, v. 3, n. 1, p. 41–60.

Patel-Schneider, P. F., Hayes, P. and Horrocks, I. (2004). OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/owl-semantics/.

Sales, T. P., Barcelos, P. P. F. and Guizzardi, G. (2012). Identification of Semantic Anti-Patterns in Ontology-Driven Conceptual Modeling via Visual Simulation. 4th International Workshop on Ontology-Driven Information Systems (ODISE 2012).

Zamborlini, V. C. (2011). Estudo de Alternativas de Mapeamento de Ontologias da Linguagem OntoUML Para OWL: Abordagens Para Representação de Informação Temporal. Federal University of Espírito Santo. Available only in Portuguese.

Zamborlini, V. C., Gonçalves, B. and Guizzardi, G. (2008). Codification and Application of a Well-Founded Heart-ECG Ontology. In Third Workshop on Ontologies and Metamodeling in Software and Data Engineering - WOMSDE 2008.

A Method to Develop Description Logic Ontologies Iteratively Based on Competency Questions: an Implementation

Yuri Malheiros[1,2], **Fred Freitas**[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brazil

[2]Departamento de Ciências Exatas – Universidade Federal da Paraíba (UFPB)
Rio Tinto – PB – Brazil

yuri@dce.ufpb.br, fred@cin.ufpe.br

Abstract. *Many methodologies and tools are proposed to improve and make easy the process of develop ontologies. We are proposing a system to develop ontologies iteratively using competency questions. The system works as follows: a user asks a question to the system, and it tries to answer the question with the knowledge encoded in an ontology. If it cannot answer correctly, the system generates new questions to ask the user for more axioms. Then, the process restarts, until the system can answer all the generated questions, including the first one. Thus, we are creating a way to define requirements, evaluate, and to add new axioms to an ontology using natural language.*

1. Introduction

Since the 90's, ontology development was more like a craft or an arcane art form than engineering, because there are no patterns to guide engineers and each development team followed its own rules [Guarino et al. 2002] [Gómez-Pérez et al. 2004]. In a clear sign of progress, systematic methodologies have been proposed to support ontology development. These methodologies address the tasks of creating and maintaining an ontology; thus, they specify an ontology lifecycle, define how to describe the ontology scope and requirements (this latter consisting of the competency questions (CQs) [Gruninger and Fox 1995]), the ontology specification itself, and its evolution, etc.

Many methodologies have been proposed to date to build an ontology, for instance, Methontology [Fernandez-Lopez et al. 1997], On-To-Knowledge [Staab et al. 2001], and Ontology 101 [Noy and McGuinness 2008], to cite but a few. They define the steps that an ontology engineer should follow to create and maintain an ontology. A noteworthy fact is that these methodologies are slightly different, but share many common features. Two important ones consist of the iterative way of development, and the use of CQs to define requirements. There are also many tools to assist ontology development. Protégé [Gennari et al. 2003], OntoStudio[1], NeOn Toolkit [del Carmen Suárez-Figueroa et al. 2008], OntoEdit [Sure et al. 2002] and WebODE [Arpírez et al. 2001] are among the most employed tools that facilitate the process of creating an ontology.

In this paper, we present a system to build ontologies from scratch or evolve an existing ontology iteratively using CQs and their respective answers. The system uses

[1]http://www.semafora-systems.com/en/products/ontostudio/

CQs written in English to check OWL DL ontologies automatically using reasoning. This is different from many works that usually check the ontologies manually or at most use SPARQL queries. The user do not need to know DL syntax or other complex language to use the system, because all the interaction is made by natural language, thus there is not barriers to nonexperts users. Furthermore, the iterative nature of the system fits in many methodologies, then it can improve well-known development processes.

The basic methodology can be described as follows: the user asks a CQ to the system, and it tries to answer the question with the knowledge encoded in an ontology. If it cannot answer correctly, a system that implements the method asks the user for some more axioms, and generates other auxiliary CQs that the user can modify and answer; then the process restarts, until it can answer all the generated CQs, including the first. We also present here a first implementation of the method, which receives CQs in natural language of various types (which are described along the article) and convert them to OWL DL.

The remainder of this paper is organized as follows: Section 2 provides a background about description logic ontologies, ontology engineering and competency questions; Section 3 presents our proposal of the system to build ontologies iteratively using competency question; Section 4 details the implementation of the system; In section 5 we show the results of tests using the system; Section 6 discusses related work; and, Section 7 concludes the paper and presents some ideas for future works.

2. Background

To set the scene of the rest of this paper, the next three sections elucidate concepts related to description logic ontologies, ontology engineering and competency questions. These concepts serve as foundation of this work.

2.1. Description Logic Ontologies

Description Logics (DLs) are a family of knowledge representation formalisms that have been gaining growing interest in the last two decades, particularly after OWL (Ontology Web Language) [Patel-Schneider et al. 2004], was approved as the W3C standard for representing the most expressive layer of the Semantic Web.

One of the most used DL languages is $\mathcal{ALC}$, due to its good trade-off between expressivity and reasoning costs. We will describe in the following since this is the language used throughout the paper. An ontology or knowledge base in $\mathcal{ALC}$ is a set of axioms a_i defined over the triple (N_C, N_R, N_O) [Baader et al. 2003], where N_C is the set of concept names or atomic concepts (unary predicate symbols), N_R is the set of role or property names (binary predicate symbols); N_O the set of individual names (constants), instances of N_C and N_R: N_{CO} is the set of classes' instances and N_{RO} the set or role instances, with $N_{CO} \cup N_{RO} = N_O$. N_C contains concepts (like Bird, Animal, etc) as well as other concept definitions as follows. If r is a role $(r \in N_R)$ and C and D are concepts $(C,\ D \in N_C)$ then the following definitions belong to the set of $\mathcal{ALC}$ concepts: (i) $C \sqcap D$ (intersection of two concepts); (ii) $C \sqcup D$ (union of two concepts); (iii) $\neg C$ (complement of a concept); (iv) $\forall r.C$ (universal restriction of a concept by a role); (v) $\exists r.C$ (existential restriction of a concept by a role); (vi) $\top$, the universal concept that subsumes all concepts, and (vii) $\bot$, the bottom concept that is subsumed by all concepts. Note that, in the definitions above, C and D can be inductively replaced by other complex concept expressions.

There are two axiom types allowed in $\mathcal{ALC}$: (i) Assertional axioms, which are concept assertions $C(a)$, or role assertions $r(a, b)$, where $C \in N_C$, $r \in N_R$, $a, b \in N_O$ and (ii) Terminological axioms, composed of any finite set of GCIs (general concept inclusion) in one of the forms $C \sqsubseteq D$ or $C \equiv D$, the latter meaning $C \sqsubseteq D$ and $D \sqsubseteq C$, C and D being concepts. An ontology or knowledge base (KB) is referred to as a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is the terminological box (or TBox) which stores terminological axioms, and $\mathcal{A}$ is the assertional box (ABox) which stores assertional axioms. $\mathcal{T}$ may contain cycles, in case at least in an axiom of the form $C \sqsubseteq D$, D can be expanded to an expression that contains C.

$\mathcal{ALC}$ semantics is formally defined in terms of interpretations, model, fixpoints, interpretation functions, etc, over a domain or discourse universe Δ [Baader et al. 2003].

2.2. Ontology engineering

According to Gómez-Perez and colleagues, ontology engineering refers to the activities related to the process, lifecycle, methods, methodologies, tools, and languages to support the ontology development [Gómez-Pérez et al. 2004]. Devedzic defines that ontology engineering covers the set of activities done during the conceptualization, design, implementation, and deployment [Devedzić 2002].

In some ways, the methodologies to develop ontologies are similar to the ones for software engineering. They provide guidance to developers and are divided in phases, for example, specification, execution, and evaluation. Besides, the process is usually iterative, and the ontology can evolve during its lifetime in a very similar way of a software, in the sense that it requires maintenance, versioning, etc. Since the early 90's, several methodologies to build ontologies have been defined, with activities like requirements definition, implementation, and evaluation.

2.3. Competency questions

Competency questions [Gruninger and Fox 1995] are a set of questions that the ontology must be capable to answer using its axioms. The questions can be used to specify the problems an ontology or a set of ontologies must solve. Thus, they work as requirements' specification of one or more ontologies. With a set of CQs at hand, it is possible to know whether an ontology was created correctly, if it contains all the necessary and sufficient axioms that correctly answer the CQs.

Many works propose the use of CQs for ontology engineering, but they usually used them to check ontologies manually, or, at most, express them as SPARQL queries. In the case of answers that arise from more complex DL reasoning, in which the answers are not present in the ontology but can be entailed by it, no other option is yet offered, but to check CQs manually, what constitutes a slow and expensive process that could be impracticable with very large ontologies or when the quantity of CQs is huge.

3. Proposal: Method to Develop Ontologies Iteratively Based on CQs

We developed a method and a system implementation to build ontologies iteratively using CQs and their respective answers. It is based on the idea of Uschold [Uschold 96], which was never tried in the Semantic Web context. Yet, all the questions and answers are written in English. The method's algorithm is given by the Figure 1:

CQOntoBuilder

Inputs: 1. CQ_{NL} (competency question in natural language) or CQ_{DL} (competency question DL)

2. $\propto \mid O \models \propto (CQ_{DL})$, i.e., the answer to CQ_{DL}

3. O = set of axioms a_i defined over the triple (N_C, N_R, N_O) [1], where

- N_C is the set of *concept names* ,
- N_R is the set of *role or property names*,
- N_I the set of *instances* of N_C and N_R:

Output: $O' \mid O' \models \propto (CQ_{DL})$

1. CQ_{DL} := conversion (CQ_{NL});
2. **If** $O \models \propto (CQ_{DL})$ **return** O;
3. **Else**
   - **a.** **{Adding new knowledge, typed by the user}**
   - **b.** $N_C' := \emptyset$;
   - **c.** **Repeat**
     - **i.** C= new Concept(read(C));
     - **ii.** $N_C' := N_C' \cup C$;
   - d. **Until user breaks;**
   - **e.** $N_R' := \emptyset$;
   - **f.** **Repeat**
     - **i.** r= new Role(read(r));
     - **ii.** $N_R' := N_R' \cup r$;
   - g. **Until user breaks;**
   - **h.** $N_I' := \emptyset$;
   - **i.** **Repeat**
     - **i.** C= new Instance(read(i));
     - **ii.** $N_I' := N_I' \cup i$;
   - j. **Until user breaks;**
   - **k.** $O' := O' \cup N_C' \cup N_R' \cup N_I'$;
   - **l.** **{Generation of a new CQ}**
   - **m.** CQ_{DL}' := generate(CQ_{DL}, O);
   - n. read ($\propto' \mid O' \models \propto' (CQ_{DL}')$);
   - **o.** **return** CQOntoBuilder ($CQ_{DL}', \propto', O'$) ;
4. **Endif**

Figure 1. Algorithm

This algorithm is recursive and receives a CQ in natural language or DL, converts it to DL when needed, and, in case it is not satisfied yet, asks for more knowledge, generates a new CQ that should help the ontology O to satisfy the original CQ and restarts this process all over again. Note that the algorithm assumes that the oracle function generate is available. For our current implementation, we assume that the user will do this job.

Example 1. An ontology with the following axioms is loaded:

$Herbivorous \equiv Animal \sqcap \forall eats.\neg meat$

$Cow \equiv Animal \sqcap \forall eats.grass$

Then, a CQ states "Are cows herbivorous?", where the expected answer is "true".

A system implementing the method tries to answer the question, but fails, because the ontology lacks the necessary axioms to infer that $Cows \sqsubseteq Herbivorous$. Next, the system generates a new CQ for the user, for instance, "are grass and meat disjoint?". If the user answers "yes" the system includes in the ontology an axiom stating that the classes Grass and Meat are disjoint ($Grass \sqsubseteq \neg Meat$). Now, the ontology has the necessary axioms to answer the initial question correctly. □

Using this iterative process, a user can evaluate if an ontology has the necessary axioms to answer questions, and can add new knowledge, "teaching" it through the answers to the CQ made by the method/system. In the current version, our system can answer many types of questions using natural language and can add new axioms to an ontology according to the answers to questions. The question generation by the system is

still being studied since it indeed represents a new DL problem, which requires additional specific research to determine for which DL languages the problem is decidable, and in case they are, the problem's computability. Currently, we are assuming that an oracle for that problem exists indeed, in this implementation, the user provides the questions.

In the next sections, we describe a first implementation of the method with its two components: the natural language query component and the ontology builder.

4. Implementation

The system includes three core components:

- Natural language query: in the system, the user can write CQs in natural language. This component parses the query, uses the knowledge specified in an ontology, and returns an answer;
- Question generator: when the system cannot answer a question, because the ontology does not have the necessary knowledge, it generates questions for the user, to gather more knowledge to answer the initial question;
- Ontology builder: all the new knowledge learned through the questions generated by the previous component are added to the ontology. This component is responsible for transforming the information of the previous component to an ontology specification language.

4.1. Natural language query

After loading an ontology, the next step of the process to build or evolve an ontology with the proposed system is to write a CQ in natural language. We choose this approach to compose a CQ, because it is easier to use natural language than description logics.

In the system, there are predefined types of questions that it understands. The types are defined by rules, and each rule is defined using grammatical tags (nouns, adjectives, verbs, etc.) and regular expression operators (*, +, ?, and |). Each word of a question is labeled using the NLP Stanford POS Tagger [Toutanova et al. 2003]. The labels are the grammatical category of the word. Then, the component verifies if the words and its POS tags match with some question rule. If it satisfies a rule, the component will perform the operations to retrieve information of the ontology according to the question type. Otherwise, the system returns that it does not understand what the user asks.

The component can find names defined in the ontology even though they are written in the question in plural, or separated by spaces, or with different capitalizations. For example, "red wine" in a question can be matched with a class "RedWine" in the ontology, or the word "cows" in a question can be matched with a class "Cow". The component tests many variations of names in the question to find the correct match in the ontology. Thus, the user can make questions in a very natural way regardless the specific notation used to specify the ontology.

This component uses OWL API [Horridge and Bechhofer 2011] and HermiT OWL reasoner [Shearer et al. 2008] to search for answers. Thus, it can infer information that is not explicitly defined in an ontology to give the correct answer.

The following are the types of questions supported. There are three simple types of CQs to check different characteristics of an ontology. We present a general explanation of each rule, usage examples, the regular expression rules and the type of answers.

4.1.1. Is-a question

The first type of question verifies if a class is subclass of another class.

Example: Is red wine a wine?
Rule: is (Noun|Adjective|Number)+ (a|an) (Noun|Adjective|Number)+
Answers: Yes, no, true or false.

Both (Noun|Adjective|Number)+ in the rule refer to classes names in the ontology. Then, this question type supports class names composed by nouns and adjectives. For example, "red wine" is a valid class name, because red is an adjective and wine a noun. The order is unimportant, thus a class name can start with a noun or an adjective. The quantity of nouns and adjectives does not matter too. The class name must have at least one word, but all combinations of nouns and adjectives with any number of words (greater than one) are possible.

4.1.2. Property value question

This type of question verifies if a property of an instance has a specified value. The system will answer "yes" or "true" if the property of the instance indeed has the specified value, and "no" or "false" in the opposite case. Property value questions have two distinct rules.

Examples: Does bancroft chardonnay have color white?
Do birds eat animals?
Rules:
(does|do) (Noun|Adjective|Number)+ have Noun (Noun|Adjective|Number)+
(does|do) (Noun|Adjective|Number)+ Verb (Noun|Adjective|Number)+
Answers: Yes, no, true or false.

In both rules the instance name is defined by the first (Noun|Adjective|Number) and the second (Noun|Adjective|Number)+ defines the value of the property. The first rule verifies only properties names starting with "has" followed by a noun. Properties like "hasColor", "hasPart", etc., are common in ontologies; therefore we created a special rule for such cases. In the second rule, the property name is a verb.

4.1.3. Existence question

The existence questions have two rules too. This type of question verifies which subclasses of a class exist. These questions support DL existential and universal restrictions. The system will answer the list of the subclasses found.

Examples: Which wines exist?
Which wines have sugar dry?
Rules:
which (Noun|Adjective|Number)+ exist
which (Noun|Adjective|Number)+ have (Noun|Verb) (some|only)? (Noun|Adjective|Number)+
Answers: A list of classes separated by commas or the word "and". For example, "red wine, white wine".

In the rules, the first (Noun|Adjective|Number)+ defines the class name. The second rule expects extra information: a property name starting with "has" followed by a noun or a verb. In the end of the second rule, there is another (Noun|Adjective|Number)+, which defines the property's value. The user can write the words "only" or "some" optionally to specify existential and universal restrictions respectively.

4.2. Ontology builder

The goal of the ontology builder component is to add new knowledge to an ontology. Answering questions generated by the system, the user acts as a teacher to the system, that stores what it learns in the ontology. The system has predefined types of questions it can generate. These questions are called system's questions (SQ), a competency question generated by the system. In this case, there are not rules for each SQ, because the system knows exactly the format of the question it will generate. The user only needs to answer the question properly.

The following are the types of SQs supported. We present a general explanation of each SQ, usage examples, axioms generated, and answers' types they expect.

4.2.1. Is-a system's question

The system uses this type of SQ when it needs to know about the subclass relation of two classes, if it is true or false.

Example: Is red wine a wine?
Answers: Yes, no, true or false.
Axiom: $RedWine \sqsubseteq Wine$

4.2.2. Property value system's question

In this type of SQ, the system looks for knowledge about the value of some property. When answering this question positively, the user specifies that a class has a certain property and that this property has a certain range of values. It is also possible to make SQs using universal and existential restrictions.

Examples: Does bancroft chardonnay have color white?
Does bird eat some grass?
Answers: Yes, no, true or false.
Axioms:
$BancrofChardonnay \sqsubseteq \forall hasColor.White$
$Bird \sqsubseteq \exists eat.Grass$

4.2.3. Existence system's question

The last type of SQ is similar to the first, but it is concerned with the multiple relation of classes and a superclass. When answering this SQ, the user is specifying that multiple classes are subclasses of one class.

Table 1. Natural language query tests with wine ontology

Competency Question	Answer
Is red wine a wine?	true
Does bancroft chardonnay have color white?	true
Which red wines exist?	Beaujolais, CabernetFranc, CabernetSauvignon, Chianti, CotesDOr, DryRedWine, Margaux, Medoc, Meritage, Merlot, Pauillac, PetiteSyrah, PinotNoir, Port, RedBordeaux, RedBurgundy, RedTableWine, StEmilion, and Zinfandel
Which wines have sugar sweet?	IceWine, LateHarvest, Port, Sauternes, and SweetRiesling

Table 2. Natural language query tests with pizza ontology

Competency Question	Answer
Is napoletana a cheesey pizza?	true
Which spicy pizzas exist?	AmericanHot, Cajun, CheeseyVegetableTopping, IceCream, PolloAdAstra, and SloppyGiuseppe
Which meaty pizza has topping some ham topping?	Capricciosa, CheeseyVegetableTopping, IceCream, LaReine, Parmense, and Siciliana

Example: Which wines exist?
Answers: A list of classes separated by commas or the word "and". For example, "red wine, white wine".
Axioms:

$$RedWine \sqsubseteq Wine$$
$$WhiteWine \sqsubseteq Wine$$

5. Results

We have some preliminary results using the components detailed in the previous section. For the natural language query component we performed three rounds of tests, each one with a different ontology. The used ontologies were the wine ontology, the pizza ontology, and the travel ontology, all available in the Protégé website[2]. For each round, we used at least one CQ for each type, except in for the pizza ontology, because it lacks individuals with properties. For the ontology builder, we test each type of SQ using the wine ontology.

5.1. Natural language query tests

In the tests of the natural language query component, we used three different ontologies. For each ontology, we show the CQs used, and the answers for them. First, we tested the wine ontology, the Table 1 displays the results using it. Further, the Table 2 has the results for the pizza ontology. Last, the Table 3 displays the results for the travel ontology.

5.2. Ontology builder

For present the results of the SQs, we created a new class in the wine ontology called TestWine. Nothing was specified about this class, only that it exists. Then, we make SQs and answer them as follow:

1. Is test wine a wine? True.
2. Does test wine have color white? True.

[2]http://protege.stanford.edu/download/ontologies.html

Table 3. Natural language query tests with travel ontology

Competency Question	Answer
Is yoga an activity?	true
Does four seasons have rating three star rating?	true
Which adventures exist?	BunjeeJumping, and Safari
Which accommodations has rating one star rating?	Campground, and Safari

3. Which test wines exist? Red wine.

For the first question, the system wrote OWL code in the ontology to define that TestWine is subclass of Wine. The code written was:

```
Class: vin:TestWine

    SubClassOf:
        vin:Wine
```

Answering the question two positively the component wrote the definition that TestWine class has the property hasColor with value White. After, the code for the Test-Wine class was:

```
Class: vin:TestWine
    SubClassOf:
        vin:Wine
        vin:hasColor only vin:White
```

For the last question, the answer specified that RedWine is subclass of TestWine. Then, the class code changed to:

```
Class: vin:RedWine

    EquivalentTo:
        vin:Wine
         and (vin:hasColor value vin:Red)

    SubClassOf:
        vin:TestWine
```

5.3. Discussion of the results

In the first CQ tested in the wine ontology, we can already see the importance of the reasoner. In the ontology, there is no code defining directly that the class RedWine is subclass of Wine. However, the system answers true. It seems correct, because RedWine is indeed a type of Wine, but if the ontology does not specify this, the answer must be false. What happened was that the system ran the HermiT reasoner before search for an answer. The reasoner infers that RedWine is subclass of Wine, then the answer is really correct. In the test of the pizza ontology, we got some strange results. For example, in the CQ "Which meaty pizza has topping some ham topping?", one of the classes listed in the answer was IceCream. It seems a wrong answer, but the ontology was created in a way that the reasoner infers this awkward relation between IceCream and MeatyPizza.

The ontology builder component is working correctly for the defined SQs. The OWL API allows the system write new axioms flawlessly. Then, we only need to extend this component to support the inclusion of more types of axioms in the future.

6. Related work

This work was inspired by the iterative way of develop ontologies proposed by many methodologies in literature, and by the use of CQs to evaluate and define requirements.

Methontology [Fernandez-Lopez et al. 1997] defines activities to perform during the ontology development and it defines the ontology life cycle too. During its life, an ontology moves through the following states: specification, conceptualization, formalization, integration, implementation, and maintenance. This life cycle seems analogous to the waterfall life cycle in software engineering [Royce 1987], however the authors make it clear that it is not an adequate path to develop an ontology. Then, it is proposed an evolving life cycle that allows the engineer to go back from any state to other if it is necessary. Thus, this life cycle permits inclusion, removal, or modification anytime during the development. The Ontology 101 methodology [Noy and McGuinness 2008] defines an iterative process. The engineer starts with a simple model and refines it during the development. The steps of this process are: determine the domain and scope, consider reusing ontologies, enumerate important terms, define the classes and the class hierarchy, define the properties, define the facets of the slots, and create instances. The authors suggest to use CQs in the first step to determine the scope of the ontology.

Other methodologies to build ontologies emerged since 1990. Lenat and Guha presented the steps of Cyc development in one of the first works in this area [Lenat and Guha 1989]. Uschold and Gruninger contribute in many papers to evolve the ontology engineering field, proposing and refining guidelines [Gruninger and Fox 1995] [Uschold and King 1995] [Uschold et al. 1996]. In 2001, the On-To-Knowledge methodology appeared, it was a result of the project with the same name [Staab et al. 2001].

During the emergence of the methodologies, many tools to support the ontology development process are proposed. The first was the OntolinguaServer in the beginning of 1990. It started only with a simple editor, and later other components were added, such an equation solver and an ontology merge tool [Farquhar et al. 1997]. The WebOnto tool was developed in 1997, its main innovation was the collaborative edition of ontologies [Domingue 1998]. Protégé, an ontology editor with extensible architecture, is one of the most popular ontology tools nowadays. This tool supports the creation of ontologies in multiple formats [Gennari et al. 2003]. In the first years of 2000, WebODE [Arpírez et al. 2001] and OntoEdit [Sure et al. 2002] appeared. The WebODE supports multiple formats of ontology, it has an editor, and components to evaluate and merge ontologies. Also, WebODE supports most of the activities and steps of Methontology. Last, the OntoEdit has similar characteristic of the previous tools, for example, extensible architecture, ontology editor, etc.

All these works presented tried to improve the way of develop ontologies. In this paper, we are not proposing a new full featured ontology editor, neither a new methodology to create ontologies, but we are creating a system to support some phases of iterative methodologies and it may be integrated in some existing tools.

7. Conclusion

In this paper, we presented our progress in developing a method and its respective system to support a novel process of DL ontology building. Two key components are already

operating, the natural language query and the ontology builder. The former is responsible to process a natural language CQs and tries to answer it using the knowledge specified in a DL ontology. The latter is concerned with incorporating new knowledge in an ontology; it uses answers stated by the users to CQs generated by the system. We also defined the process to build or evolve an ontology iteratively using the system.

There are still limitations in the work. Each component needs to evolve. The natural language query component must support more types of question and treat more intrinsic details of the written language. The ontology builder needs to support more SQs too. Finally, we need to study the problem and develop the automatic question generation when some knowledge is missing in the ontology and the system cannot answer a question correctly. For now, this process is made manually.

As future work, besides evolving the system's components, we intend to build a complete tool for ontology engineers based on the proposed method. Then, they can create or evolve an ontology using all the process defined in this paper. The tool may be integrated with popular ontology environments like Protégé and NeOn.

References

Arpírez, J. C., Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2001). Webode: a scalable workbench for ontological engineering. In *Proceedings of the 1st international conference on Knowledge capture*, K-CAP '01, pages 6–13, New York, NY, USA. ACM.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, New York, NY, USA.

del Carmen Suárez-Figueroa, M., de Cea, G. A., Buil, C., Dellschaft, K., Fernández-López, M., García, A., Gómez-Pérez, A., Herrero, G., Montiel-Ponsoda, E., Sabou, M., Villazon-Terrazas, B., and Yufei, Z. (2008). D5.4.1 neon methodology for building contextualized ontology networks.

Devedzić, V. (2002). Understanding ontological engineering. *Commun. ACM*, 45(4):136–144.

Domingue, J. (1998). Tadzebao and webonto: Discussing, browsing, and editing ontologies on the web. In *In Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop*.

Farquhar, A., Fikes, R., and Rice, J. (1997). The ontolingua server: a tool for collaborative ontology construction. *Int. J. Hum.-Comput. Stud.*, 46(6):707–727.

Fernandez-Lopez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA.

Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., and Tu, S. W. (2003). The evolution of protege: an environment for knowledge-based systems development. *Int. J. Hum.-Comput. Stud.*, 58(1):89–123.

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2004). *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web*. Advanced Information and Knowledge Processing. Springer.

Gruninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies.

Guarino, N., Welty, C., and Common, E. (2002). Evaluating ontological decisions with ontoclean.

Horridge, M. and Bechhofer, S. (2011). The owl api: A java api for owl ontologies. *Semant. web*, 2(1):11–21.

Lenat, D. B. and Guha, R. V. (1989). *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Noy, N. F. and McGuinness, D. L. (2008). Ontology development 101: A guide to creating your first ontology.

Patel-Schneider, P. F., Hayes, P., and Horrocks, I. (2004). OWL web ontology language semantics and abstract syntax. W3C recommendation, W3C. Published online on February 10th, 2004 at `http://www.w3.org/TR/2004/REC-owl-semantics-20040210/`.

Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, ICSE '87, pages 328–338, Los Alamitos, CA, USA. IEEE Computer Society Press.

Shearer, R., Motik, B., and Horrocks, I. (2008). HermiT: A Highly-Efficient OWL Reasoner. In Ruttenberg, A., Sattler, U., and Dolbear, C., editors, *Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 EU)*, Karlsruhe, Germany.

Staab, S., Studer, R., Schnurr, H.-P., and Sure, Y. (2001). Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26–34.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., and Wenke, D. (2002). Ontoedit: Collaborative ontology development for the semantic web. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 221–235, London, UK, UK. Springer-Verlag.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Uschold, M., Gruninger, M., Uschold, M., and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136.

Uschold, M. and King, M. (1995). Towards a methodology for building ontologies. In *In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*.

Unifying Phenotypes to Support Semantic Descriptions

Eduardo Miranda [1], **André Santanchè** [1]

[1]Institute of Computing – State University of Campinas
Av. Albert Einstein, 1251 – Cidade Universitária, Campinas, Brazil

eduardo.miranda@students.ic.unicamp.br, santanche@ic.unicamp.br

Abstract. *In life sciences, there are several biological datasets shared through the web. All this abundance of data carries a great opportunity to explore complex relationships among the diversity of species. However, their physical format varies from independent data files to databases, which are heterogeneous in model and representation, hampering their integration. Ontologies are one of the promising choices to address this challenge. However, the existing digital phenotypic descriptions are stored in semi-structured formats, making extensive use of natural language. If on one hand, this patrimony is highly relevant, on the other hand, converting it in ontologies is not a straightforward task. The present article addresses this problem adding an intermediate step between semi-structured phenotypic descriptions and ontologies. It remodels semi-structured descriptions to a graph abstraction in which the data are linked. Graph transformations subsidize the transition from semi-structured data representation to a more formalized representation through ontologies.*

1. Introduction

Bioinformatics is the science of integrating, managing, mining and interpreting information from biological data [Gibas and Jambeck 2001]. In the life science field, there are a large number of distributed biological datasets freely available and ready to use. However, this wealth of information has hardly been tapped even today due its distributed nature, heterogeneity and complex data types and representation [Parr et al. 2012]. In this scenario, their combination and interconnection are barely feasible [Quan 2007]. A massive amount of relevant information is hidden in the potential connection of unrelated files.

In this work we are interested in a specific biology context, in which biologists apply computational tools to build and share digital descriptions of living beings as phenotypes. These descriptions are a fundamental starting point for several biology tasks, like living beings identification and tools for phylogenetic tree analysis. Even though the last generation of these tools is based on open standards (e.g., XML), the descriptions are still based on textual sentences in natural language [Balhoff et al. 2010].

Semantic integration in this context is one of the main challenges. Besides ontologies to support phenotype description, there are tools to annotate descriptions by associating ontology concepts to textual descriptions [Balhoff et al. 2010]. This distinction between description and their annotations based on ontologies does not consider that descriptions can conversely contribute to ontology expansion and revision. The challenge in this work is to establish a model to represent a common denominator among phenotipical description standards, which will support findings in the latent semantics implicit in relations in a strategy inspired by folksonomies. These semantics can guide the interaction between textual descriptions and ontologies.

In a previous work [Alves and Santanchè 2013], we showed that the latent semantics presented in tags and their correlations, as a product of an organic work collectively produced by a community on the web (the folksonomies), can be exploited to expand and review ontologies. While the model behind folksonomies is based on the correlation of three elements – tags, resources and users – descriptions in the biological context present a more complex and specialized structures. Co-occurrence is a strong principle we considered to extract latent semantics. The main idea is that the set of tags put together in a given resource can provide a "context" to interpret each tag. Consider a tag *cell*, which can have a distinct interpretation according to the context. The co-occurrence with the tags *cytoplasm* or *organelle* will put it in the biology context. Moreover, the compilation of data concerning the occurrence and co-occurrence of millions of tags can support the analysis of similarity among terms – see more details in [Alves and Santanchè 2013]. We consider that we can apply an equivalent technique to put terms of phenotype descriptions in a context, to improve their interpretation and correlation.

The present paper addresses this problem in exploiting existing biology assets related to phenotypic descriptions, and the latent semantics resulting from their interconnection, to support their development towards a richer semantical representation, as part of ontologies. It implies promoting relations among concepts to first class citizens. Accordingly, we designed a three layered method illustrated in Figure 1, in which graph databases intermediate this evolvement process from fragmentary data sources to accomplish full integration descriptions as ontologies.

Our approach remodels semi-structured descriptions to a graph abstraction, in which the data can be integrated more easily. Graph transformations are applied for the transition from a semi-structured data representation to a more formalized representation through ontologies. As we will further explain, this graph representation will also support an analytical tool to compare data across studies, wherein it will help evolutionary biologists to answer evolutionary questions. This paper presents a work in progress concerning the first step of this method, focusing in the integration of data from the semi-structured data layer and their transition to the graph data abstraction layer. Our proposed graph-based model is derived from a comparative analysis among four standards related to phenotype description, plus a practical experiment.

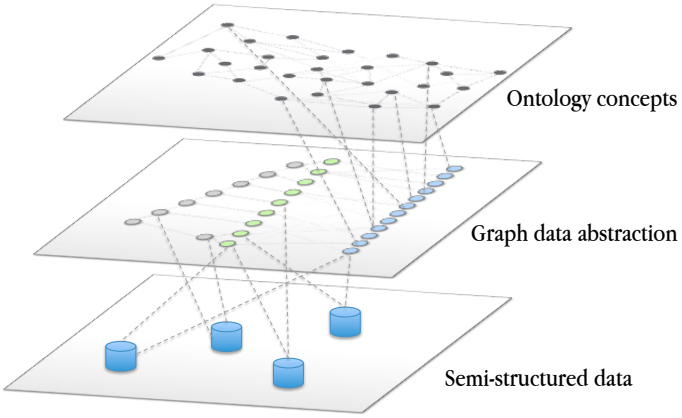


Figure 1. Three layers method diagram.

This paper is organized as follows: Section 2 summarizes the related work; Section 3 presents the comparative analysis which subsidizes our minimal common denominator model; Section 4 presents out graph-based model; Section 5 shows a practical experiment of unifying phenotypes; Section 6 presents concluding remarks.

2. Related Work

Integration is a key point as humans are progressively unable of handling the sheer volume of data presented [Bell et al. 2009]. It is an important step towards knowledge discovery [Lenzerini 2002]. The integration of digital phenotype descriptions is a relevant challenge in this context since they support fundamental biology tasks as the building of identification keys for living beings and can support the creation of a complete evolutionary Tree of Life [Parr et al. 2012] assembling genomic and morphological data so as to congregate the phylogenetic relationships among all living or extinct organisms [Ciccarelli et al. 2006]. Likewise, integrating these data may contribute to better understanding of how a morphological trait became organized and evolved over time [Mabee 2006].

Recent approaches enrich descriptions via ontology annotations, using the Entity-Quality (EQ) formalism for phenotype modeling. EQ is a representation [Balhoff et al. 2010] which associates ontology entity terms (E) – e.g., bone or vertebra from Teleost Anatomy Ontology (TAO) – with quality terms (Q) – e.g., triangular, horizontal, smooth from the Phenotype and Trait Ontology (PATO) [Dahdul et al. 2010]. Ontologies have gained wide acceptance in biology due to their ability of representing knowledge and also the advantage of querying and reasoning information [Gkoutos et al. 2004]. Furthermore, semantic web standards to represent ontology concepts with unique identifiers facilitates interoperability across databases [Mabee et al. 2007]. Recently, several tools have emerged to support annotation of biological phenotypes using ontologies, e.g., Phenex (*http://phenoscape.org/wiki/Phenex*) and Phenote (*http://www.phenote.org/*), both curation tools designed for annotation of phenotypic characters with ontology concepts using EQ formalism [Balhoff et al. 2010].

[Dahdul et al. 2010] developed a workflow for curation of phenotypic characters extracted from scientific publications. It is important to note the limitations of this curation process, considering that it is very time-consuming since it is manually carried out by domain experts.

3. Common Denominator

There is a wide variety of representation formats for phenotype description, adopted by information systems and open standards, which represent differently the same information. In this section, we analyze four of them – Xper^2, SDD, Nexus and NeXML – looking for a minimal common denominator, which is the foundation for our graph-based model, to be used to link related information.

SDD, Nexus and NeXML are widely adopted open standards further detailed. Xper^2 (*http://lis-upmc.snv.jussieu.fr/lis/*) is a management system adopted by the systematist community, for the storing, editing and analyzing of phenotype descriptive data. It focuses mainly on taxonomic descriptions, allowing creation, sharing and comparison of identification keys [Ung et al. 2010a, Ung et al. 2010b]. Xper^2 was developed in the Laboratoire Informatique & Systématique of the University Pierre et Marie Curie and

this work is part of a bigger project in collaboration with this lab. Therefore, Xper^2 was adopted for our practical experiments.

In order to illustrate our analysis, let us consider a practical case, in which a biologist is building a phenotype description of monitor lizards (genus *Varanus*). The process starts with the biologist collecting observations of lizards, organized as characters and character states (C, CS). [Pimentcl and Riggins 1987] defined character as *"a feature of organisms that can be evaluated as a variable with two or more mutually exclusive and ordered states"*. The observations involved the species *Varanus albiguralis* and *Varanus brevicauda*. The final result is the character-by-taxon matrix illustrated in Figure 2.

	nostrils' form	transversal section of the tail	nuchal scales
Varanus albiguralis	2	1	2
Varanus brevicauda	1	2	1

Nostrils' form
1 – well round
2 – oval or split-like

Transversal section of the tail
1 – laterally compressed
2 – roundish

Nuchal scales
1 – same size than head scales
2 – bigger than head scales

Figure 2. Character-by-taxon matrix

In order to transform these observations to digital records and generalize them – e.g., devising general characters and states observed in a genre of monitor lizards – the biologist will use a tool like Xper^2. Phenotypes descriptions can be stored in the Xper^2 native format or can be exported to the SDD open format. The Structure Descriptive Data (SDD) (*http://wiki.tdwg.org/SDD*) is a platform and application-independent XML-based standard developed by the Biodiversity Information Standards (historic acronym: TDWG) for recording and exchanging descriptions of biological and biodiversity data of any type [Hagedorn 2007]. SDD is adopted by several other phenotype description tools – e.g., Lucid Central (*http://www.lucidcentral.org*) and Linnaeus II (*http://www.eti.uva.nl/*).

We further introduce some key elements of the SDD format, which are recurrent in the formats confronted in this section. A SDD description comprises, in a single file, a domain schema and its instances. Figure 3 shows a diagram with a fragment of a SDD file containing the description of a varanus lizard. A (C,CS) description in SDD has two main blocks: (i) defines the characters involved and their possible states – Figure 3 top; (ii) describes an Operational Taxonomic Unit (OTU) using the characters defined in (i) – Figure 3 bottom. OTU is a biology term which refers to a given entity in sampling level adopted to the study – e.g., a specimen, a gender etc.

<*CategoricalCharacter*>s and their <*States*> (shown in Figure 3 top) are primitives to describe an OTU [Hagedorn 2007]. Each <*CategoricalCharacter*> has its <*Representation*> – comprising a label and a description as plain texts – and a set of <*StateDefinition*> elements with their possible states. <*CategoricalCharacter*> and <*StateDefinition*> elements defined here will be referred throughout the XML document by their ids.

The <*CodedDescription*> (Figure 3 bottom) links the OTU being described to *States* of each <*CategoricalCharacter*>. It has two essential items: (i) the OTU

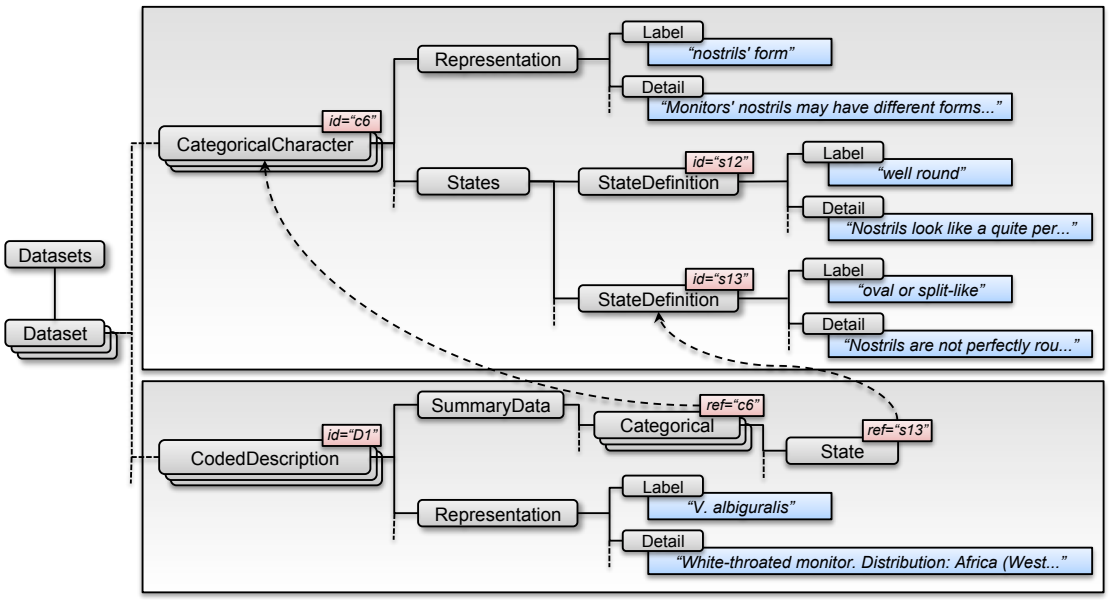


Figure 3. Fragment of SDD Schema with Instances [1]

being described, where its name and description are listed in natural language under <*Representation*>; (ii) a set of character and values (<*Categorical*> and <*State*>), which address the characters defined in the previous section through the *ref* attribute. It is possible and usual to define multiple states for a character of a given OTU. A first integration, problem observed here is that each character or OTU described does not have a global unique identification among documents. Therefore, the description can only be used by the document where it was declared and it is not possible to guarantee the equivalence of two or more <*CategoricalCharacters*>.

In Figure 5 we expand our analysis to the Xper^2 native format, Nexus and NeXML. Our study addresses mainly morphological character descriptions. Figure 5 provides simplified diagrams focusing on the elements to record descriptions, which will be confronted here. Figure 4 presents the symbols adopted in the diagram. All the formats adopt XML and the symbols represent the relations among elements and their respective cardinality. Five types of elements, which are focus of our analysis, receive special symbols: the `Entity` being described, which can be a taxon or a specimen; the `Character` definition and its respective association with entities (`Character instance`); the `State` definition and its respective association with entities (`State instance`).

Nexus [Maddison et al. 1997] is an extensively used file format developed for storage and exchange of phylogenetic data, including morphological and molecular characters, taxa distances, genetic codes, phylogenetic trees etc. It was designed in 1987 and it is still used by many popular software as Xper^2 (*http://lis-upmc.snv.jussieu.fr/lis/*), Mesquite (*http://mesquiteproject.org/*), MrBayes (*http://mrbayes.sourceforge.net/*) and data repositories, like TreeBASE(*http://treebase.org/*) and Dryad (*http://datadryad.org/*). Nexus gathers together (C,CS) based descriptions and related trees [Vos et al. 2012].

[1] Knowledge base of the genus Varanus from *http://lis-upmc.snv.jussieu.fr/xper2/infosXper2Bases/liste-bases-recherche.php*

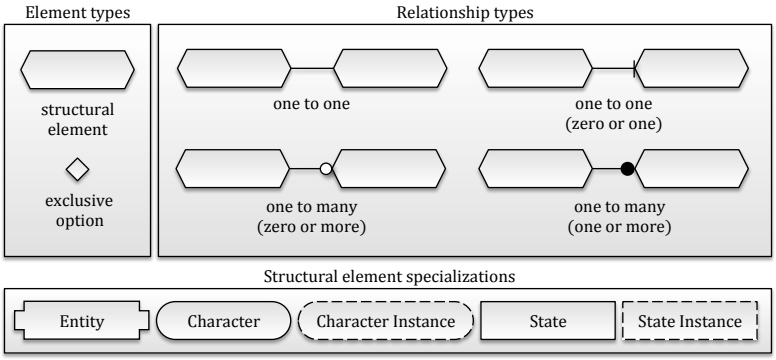


Figure 4. Symbols and semantic used in the diagrams

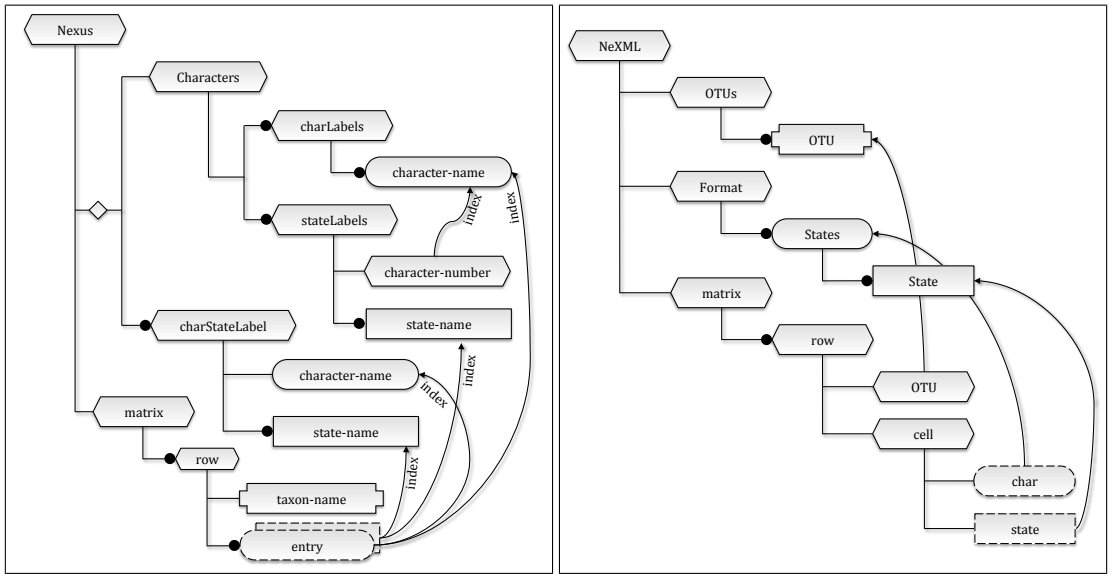
NeXML (*http://www.nexml.org*) [Vos et al. 2012] is a standard inspired by the Nexus. It supports and extends Nexus functionalities and addresses some Nexus limitations – e.g., connects objects with ontology concepts, supports citations and annotations [Vos et al. 2012]. In order to accomplish full compatibility and interoperability among different environments, NeXML defines a formalized XSD grammar and enables semantic annotations of any element in a NeXML document, which goes towards to a "Minimum Information About a Phylogenetic Analysis" (MIAPA) standard.

These comparative diagrams show that even if the structures are arranged differently, they address the same key elements. All formats organize data in accordance with the (C,CS) data model that, in practice, is an entity-attribute-value (EAV) model, in which entities are OTUs, attributes are characters and values are character-states [Vos et al. 2012]. Nexus and NeXML formats define a matrix, in which OTUs are listed in rows, characters are columns and the cells contain a numeric code for a specific character-state (see Figure 2). Although Xper^2 and SDD do not define a matrix, both formats have a similar structure to describe OTUs with their (C, CS) records.

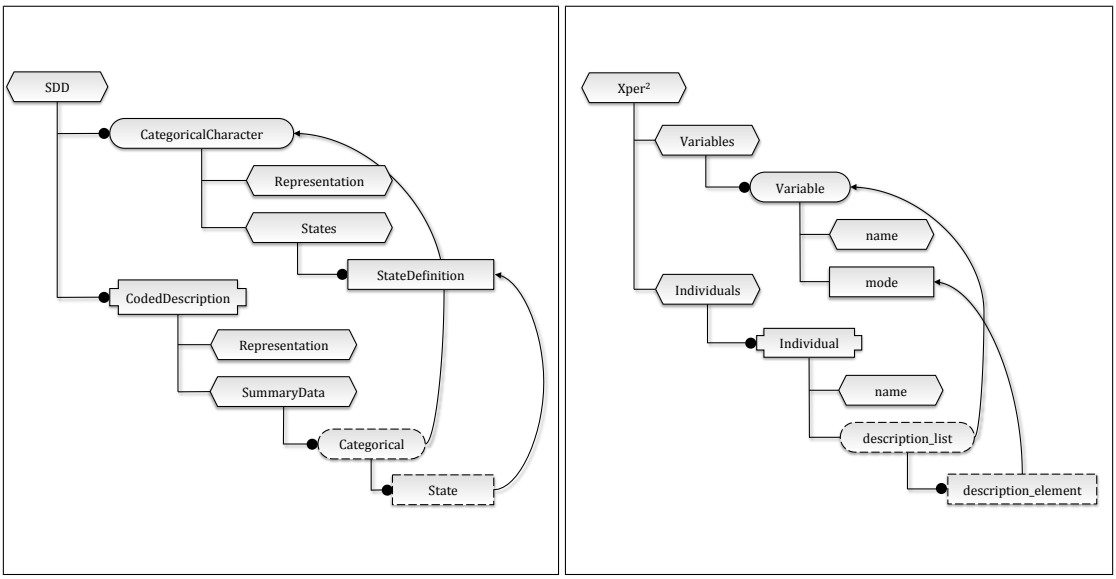
4. From XML Structures to Graphs

The next step in our Three Tier Method is designing a graph model. In a previous work [Alves and Santanchè 2013], we have compared several approaches to capture latent relations+semantics among tags produced collaboratively. Graph models to represent and analyze data were a common denominator. The role of the graph is not to reflect all details of the original model. The central challenge is how to abstract key elements, for which we are looking for potential relations to be discovered. It is a movement from the latent semantics to an explicit semantics expressed as links.

On one hand, we devised in the previous section the common denominator we are looking for: OTUs, character and character states. On the other hand, a second important ingredient is devising what is our target in ontologies. As mentioned in Section 2, a predominant ontology model for phenotype descriptions is the Entity-Quality (EQ) [Balhoff et al. 2010]. An Entity refers to the "part" of the OTU being described, which is related to one or more Qualities. In a comparison with the (C, CS) approach, a Character comprises an Entity plus the Quality involved in the description in a single textual sentence. A State is a complementary part of the Quality. Even though it is not a trivial task to split Characters into their components of Entity and Quality, a first step will



(a) *Nexus* (b) *NeXML*



(c) *SDD* (d) *Xper*^{2}

Figure 5. Formats for representing phylogenetic data

be linking disperse elements referring to the same semantic concept.

Departing from the key elements identified in the previous section, we can devise the following linking discovery challenges:

- Which OTUs in the graph refer to the same real world OTU (link OTU-OTU)?
- Which characters can be applied to each OTU (link OTU-character)?
- Which states for each character can be observed in each OTU (link OTU-character-state)? Conversely, which OTUs have a given character+state?

The answer to these questions will enable to integrate, summarize and compare data concerning each OTU and each character. Therefore, it becomes possible to answer queries like:

- What are the possible colors of a Varanus tongue?
- Which animals present an oval nostrils form?

The discovery process is carried by graph transformations. As graphs are crucial for our modeling approach, our method was built over graph databases. These databases reduce the gap between how data is modeled (as graphs) and how it is stored. It is capable of representing data structures with high abidance. Compared with relational databases, graph databases do not require join operations because it is done implicitly traversing the graph from node to node. Graph databases are less schema-dependent and for this reason, they can scale more easily in size and complexity as the application evolves.

The questions stated before were the basis to conceive the model presented in Figure 6. We adopted the *property graph* model, in which nodes and relationships can maintain extra metadata as a set of key/value pairs. Moreover, relationships are typed, enabling to create multi-relational networks with heterogeneous sets of edges. Different from single-relational networks, in which edges are of the same type, multi-relational networks are more appropriate to represent complex domain models, due the variety of relationship types in the same graph [Rodriguez and Shinavier 2010].

In our graph model, OTUs and character-states are nodes connected by characters (edges). Therefore the statement *"V. albiguralis has a well round tail shape"* becomes V. albiguralis (node) $\rightarrow$ tail shape (edge) $\rightarrow$ well round (node).

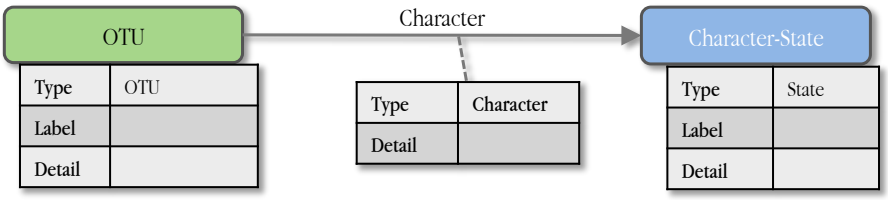


Figure 6. ***Property graph model to represent phenotype descriptions.***

5. Practical Experiment of Unifying Phenotypes

We have implemented an automatic process to ingest SDD files into a graph database, in order to show the linking possibilities raised by our model. In our experiments, we use the Neo4j (*http://www.neo4j.org/*), an open-source graph database. Our data integration processing flow is divided into the main stages: preprocessing, data ingestion, data linkage.

One of the problems faced in bioinformatics is related to the identification of objects within and across repositories [Page 2008]. More precisely, an object may refer to a taxon, gene, anatomical feature, phenotypic description, geographical location etc. Uniquely identifying those objects is undoubtedly a key point for the success of our proposed solution.

In order to address this issue, some organizations – e.g., Universal Biological Indexer and Organizer (uBio), Integrated Taxonomic Information System (ITIS), Catalogue of Life (CoL), The International Plant Names Index (IPNI), National Center for Biotechnology Information (NCBI) etc. – incorporated into their projects the Life Science Identifiers (LSIDs), which was proposed by the Object Management Group (OMG)

(*http://www.omg.org/*). LSID is a persistent, location-independent resource identifier, whose purpose is to uniquely identify biological resources [Clark et al. 2004]. The persistent property refers to the fact that LSID identifiers are unique, can be assigned to only one object forever and they never expire. The location-independent property specifies that each authority locally creates LSIDs and they are the responsible to guaranteeing the uniqueness of LSIDs.

We applied LSIDs to unify OTUs in the graph referring to the same real world object. In order to find a valid LSID, we adopted the Global Names Resolver (GNR) web service (*http://resolver.globalnames.org/*) that executes exact or fuzzy matching against canonical forms of scientific names in 170 distinct data sources. The Canonical form (cf) is the simplest, most complete and unambiguous form of a name. The Canonical form of scientific names consists of the genus and species – when applied – with no authorship, rank, nomenclatural annotation or subgenus.

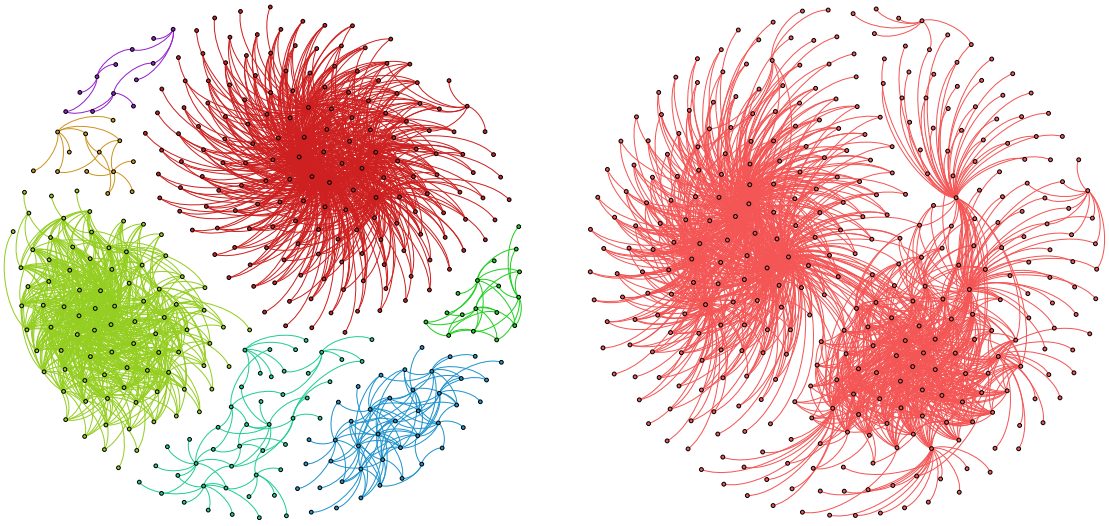
Our system used three of the six types of matching offered by the GNR resolver: (i) exact matching; (ii) exact matching of canonical forms – this process reduce a given name to its canonical form and checks it with an exact match; (iii) fuzzy matching of canonical forms – uses a modified version of the TaxaMatch algorithm [Rees 2008] and it intends to work around misspellings errors. It does a fuzzy match of the canonical form of a given name – even with mistakes – against spellings considered correct. The GNR resolver reports the matching quality (*"confidence score"*) for each match.

The matching module of the system is still a work in progress, but we already have obtained some relevant results to show the viability of our approach. From the LIS knowledge base we collected 7 distinct morphological descriptions: genus *Varanus*; species *Varanus gouldii*, *Varanus timorensis*, *Varanus auffenbergi* and *Varanus scalaris*; species groups *Varanus indicus*, *Varanus prasinus*, *Varanus salvator*; and Autralian spiny-tailed monitor lizards. Through Xper^2 those morphological descriptions were exported to the SDD format and imported into the graph database, with no preprocessing. Figure 7(a) shows an overview of the resulting graph without labels. We can note the disconnectedness of the graph (7-partite graph). On the other hand, Figure 7(b) shows the same knowledge after employing the LSID unification. The graphs became connected. Before applying the LSID unification the graph had 74 distinct taxonomic units (TUs). After performing the LSID unification its total reduced to 44 TUs, i.e., 30 taxonomic units (40%) were recurring and were integrated in a single node.

The next step is to link equivalent characters of the same OTU, enabling integration of states of the same character. In the present stage of this research we apply a simple matching algorithm. One example of our preliminary results is presented in the diagram of Figure 8. As can be seen, our algorithm was able to unify all "nuchal scales" characters, by defining the same type to the edges. Moreover, we unified and congregated the possible states observed for this character across different description files.

6. Conclusion

Several initiatives propose to relate phenotype descriptions with ontologies to enable a semantic integration. The challenge is how to expand and revise the ontology while new descriptions were created. Tools which annotate descriptions with ontologies address them as an external artifact crafted apart, disregarding the synergy between building an ontol-



(a) *Graph 7-partite* (b) *Connected graph*

Figure 7. Varanus knowledge base

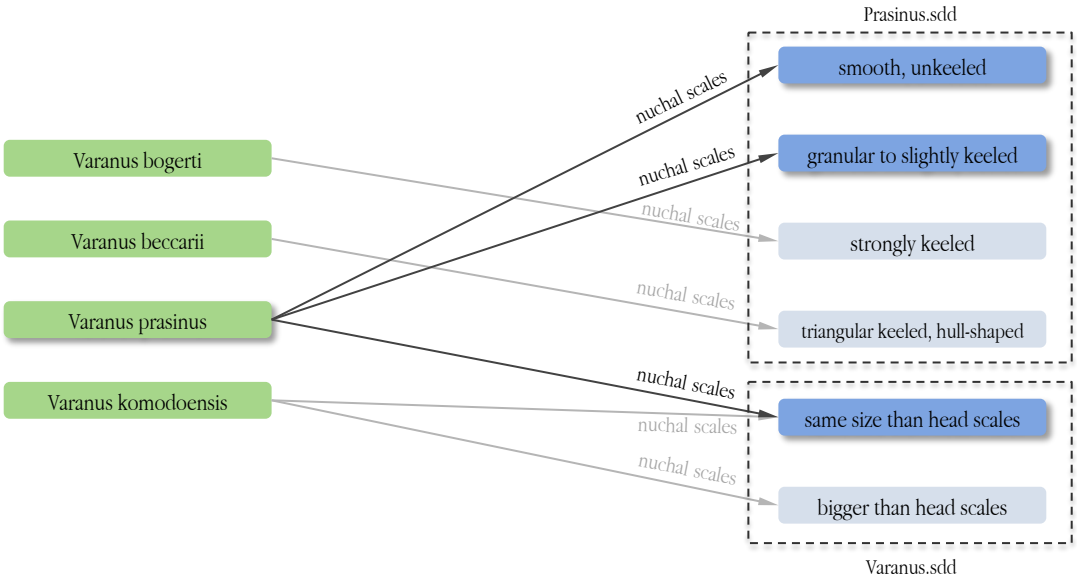


Figure 8. Graph Diagram

ogy and using it. [Shirky 2005] emphasizes the importance of the semantics organically built by a community, where a binary categorization approach – in which a concept A "is" or "is not" part of a category B – to a probabilistic approach – in which a percentage of people relates A to B. This work contributes in this direction. Inspired by previous work, which explores latent semantics in folksonomies, this work analyzes standards to describe phenotypes to find a common denominator, which is the bases to link descriptions.

The main contribution of this work is to create the basis to exploit the latent semantics in the descriptions. The viability and the potential of our approach were tested by experiments. These experiments are the first steps to exploit a bigger latent semantics scenario. Moreover, having the capability of integrating knowledge around taxonomic units

will enable, for instance, evolutionary biologists to generate new research questions, gain predictive insight or confront evolutionary hypotheses. More complete answers might be provided as new data sources are integrated.

Our representation in a graph database is aligned with the RDF [Manola and Miller 2004] graph-based representation, which will be the next step to achieve the third layer. The challenge will be to map labels of character/character-states in RDF properties/values. The unification of characters and states, as shown on this preliminary work, is a first and high relevant step for this mapping. Since several ontologies related to phenotype descriptions are in OWL, the relations discovered in our graph can subsidize a better matching of labels and concepts in OWL ontologies by confronting relations. For example, to enhance the match of a character label (in the graph database) with an OWL property, it is possible to consider the states allowed by the character, confronting them with the property range (values allowed by the property).

There are several possible ways to extend this work. One possible way is to incorporate morphological descriptions stored in other knowledge bases, e.g., MorphoBank (*http://morphobank.org/*) or Dryad (*http://datadryad.org/*). Another direction is to investigate correlations between *State* nodes and ontology terms.

Acknowledgment

Work partially financed by (CNPq 138197/2011-3), the Microsoft Research FAPESP Virtual Institute (NavScales project), CNPq (MuZOO Project and PRONEX-FAPESP), INCT in Web Science(CNPq 557.128/2009-9) and CAPES, as well as individual grants from CNPq.

References

Alves, H. and Santanchè, A. (2013). Folksonomized Ontology and the 3E Steps Technique to Support Ontology Evolvement. *Journal of Web Semantics*, 18(1):19–30.

Balhoff, J. P., Dahdul, W. M., Kothari, C. R., Lapp, H., Lundberg, J. G., Mabee, P., Midford, P. E., Westerfield, M., and Vision, T. J. (2010). Phenex: Ontological annotation of phenotypic diversity. *PLoS ONE*, 5(5):e10500.

Bell, G., Hey, T., and Szalay, A. (2009). Beyond the data deluge. *Science*, 323(5919):1297–1298.

Ciccarelli, F. D., Doerks, T., Von Mering, C., Creevey, C. J., Snel, B., and Bork, P. (2006). Toward automatic reconstruction of a highly resolved tree of life. *Science*, 311(5765):1283–1287.

Clark, T., Martin, S., and Liefeld, T. (2004). Globally distributed object identification for biological knowledgebases. *Briefings in bioinformatics*, 5(1):59–70.

Dahdul, W. M., Balhoff, J. P., Engeman, J., Grande, T., Hilton, E. J., Kothari, C., Lapp, H., Lundberg, J. G., Midford, P. E., Vision, T. J., Westerfield, M., and Mabee, P. M. (2010). Evolutionary characters, phenotypes and ontologies: Curating data from the systematic biology literature. *PLoS ONE*, 5(5):e10708.

Gibas, C. and Jambeck, P. (2001). *Developing bioinformatics computer skills*. O'Reilly Media, Inc.

Gkoutos, G., Green, E., Mallon, A.-M., Hancock, J., and Davidson, D. (2004). Using ontologies to describe mouse phenotypes. *Genome Biology*, 6(1):R8.

Hagedorn, G. (2007). *Structuring Descriptive Data of Organisms – Requirement Analysis and Information Models*. PhD thesis, Universität Bayreuth,Fakultät für Biologie, Chemie und Geowissenschaften.

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM.

Mabee, P. M. (2006). Integrating evolution and development: the need for bioinformatics in evo-devo. *BioScience*, 56(4):301–309.

Mabee, P. M., Ashburner, M., Cronk, Q., Gkoutos, G. V., Haendel, M., Segerdell, E., Mungall, C., and Westerfield, M. (2007). Phenotype ontologies: the bridge between genomics and evolution. *Trends in ecology & evolution*, 22(7):345–350.

Maddison, D. R., Swofford, D. L., and Maddison, W. P. (1997). Nexus: An extensible file format for systematic information. *Systematic Biology*, 46(4):590–621.

Manola, F. and Miller, E. (2004). RDF Primer – W3C Recommendation. Technical report, W3C.

Page, R. (2008). Biodiversity informatics: the challenge of linking data and the role of shared identifiers. *Briefings in Bioinformatics*, 9(5):345–354.

Parr, C. S., Guralnick, R., Cellinese, N., and Page, R. D. (2012). Evolutionary informatics: unifying knowledge about the diversity of life. *Trends in ecology & evolution*, 27(2):94–103.

Pimentcl, R. A. and Riggins, R. (1987). The nature of cladistic data. *Cladistics*, 3(3):201–209.

Quan, D. (2007). Improving life sciences information retrieval using semantic web technology. *Briefings in bioinformatics*, 8(3):172–182.

Rees, T. (2008). Taxamatch, a "fuzzy" matching algorithm for taxon names, and potential applications in taxonomic databases. In Weitzman, A. and Belbin, L., editors, *Provisional Abstracts of the 2008 Annual Conference of the Taxonomic Databases Working Group*, Fremantle, Australia. Biodiversity Information Standards (TDWG) and the Missouri Botanical Garden.

Rodriguez, M. A. and Shinavier, J. (2010). Exposing multi-relational networks to single-relational network analysis algorithms. *Journal of Informetrics*, 4(1):29 – 41.

Ung, V., Causse, F., and Vignes Lebbe, R. (2010a). Xper^2: managing descriptive data from their collection to e-monographs.

Ung, V., Dubus, G., Zaragüeta-Bagils, R., and Vignes-Lebbe, R. (2010b). Xper2: introducing e-taxonomy. *Bioinformatics*, 26(5):703–704.

Vos, R. A., Balhoff, J. P., Caravas, J. A., Holder, M. T., Lapp, H., Maddison, W. P., Midford, P. E., Priyam, A., Sukumaran, J., Xia, X., et al. (2012). Nexml: rich, extensible, and verifiable representation of comparative data and metadata. *Systematic Biology*, 61(4):675–689.

Part II

Short Papers

Um Estudo de Caso na Construção de Ontologias Biomédicas: uma Ontologia de Domínio sobre Hemoterapia

Fabrício M. Mendonça[1], Maurício B. Almeida[1]

[1]Escola de Ciência da Informação – Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 - Campus Pampulha – 31.270-901 – Belo Horizonte – Brasil

fabriciommendonca@gmail.com, mba@eci.ufmg.br

Resumo. *O uso de ontologias possibilita organizar, explicitar e compartilhar o conhecimento de um dado domínio. O presente artigo descreve um estudo de caso no domínio da hemoterapia, em que se tem construído uma ontologia sobre componentes do sangue humano. Como pesquisa em andamento, apresentam-se os resultados parciais obtidos com a ontologia proposta.*

Abstract. *The use of ontologies enables one to organize, to make explicit and to share knowledge about a certain domain. This article describes a case study in the domain of blood transfusion, in which we have developed an ontology about human blood components. As an ongoing research, we present the partial results obtained with the proposed ontology.*

1. Introdução

Nos últimos anos, ontologias têm sido amplamente utilizadas na descrição formal do conhecimento científico para uso em sistemas de informação. O uso de ontologias representa uma evolução nas práticas atuais de modelagem para sistemas de informação, uma vez que possibilitam explicitar o conhecimento de um domínio, promovem o compartilhamento do conhecimento e favorecem a integração da informação [Guarino 1998], [Wand e Weber 2004].

Nesse contexto, o presente artigo descreve um estudo de caso no domínio do sangue humano que envolve a construção de uma ontologia sobre hemocomponentes e hemoderivados, denominada de HEMONTO no âmbito de um projeto biomédico de escopo maior denominado de *Blood Project* [Almeida et al. 2011]. Planeja-se que a ontologia resultante funcione como um repositório de conhecimento científico, sendo utilizada, por exemplo, como instrumento para anotação e, nesse sentido, preencher a lacuna causada pela falta de uma representação formal voltada para os hemoderivados do sangue e pelas possibilidades limitadas de recuperação da informação dessa área em função do uso de ferramentas gerais. Além disso, espera-se que a ontologia proposta possa facilitar as atividades de modelagem ou avaliação dos sistemas de informação no domínio tratado, podendo assim ser classificada como uma *ontology-driven information systems* [Guarino 1998], baseada em um conjunto de princípios que se convencionou chamar de realismo ontológico [Smith and Ceusters 2010].

O restante do presente artigo está estruturado da seguinte forma: a seção 2 apresenta a metodologia adotada para a construção da ontologia; na seção 3, apresenta-se o conteúdo parcial da HEMONTO; e, na seção 4, são feitas as considerações finais.

2. Metodologia de construção da ontologia proposta

A construção de tal ontologia HEMONTO foi conduzida da seguinte forma: (i) aquisição de conhecimento no domínio em estudo através de materiais de referência; (ii) reaproveitamento do conteúdo de outros glossários, padrões e outras ontologias; e (iii) utilização de alguns artefatos de representação para explicitar o conhecimento representado pela ontologia. Os passos (i) e (ii) estão descritos na subseção 2.1 e o passo (iii) é descrito na subseção 2.2.

2.1 Ontologias e materiais de referência

Para o estudo do domínio tratado, foi selecionado inicialmente guia para o uso de hemocomponentes [Brasil 2008]. O documento aborda diretrizes para manipulação de hemocomponentes, funcionando como um instrumento de apoio para a prescrição médica referente à escolha do hemocomponente mais adequado para transfusão.

Paralelamente ao estudo do domínio tratado, procedeu-se com uma revisão de literatura de ontologias relevantes para o domínio do sangue e domínios correlatos, uma vez que se desejava reaproveitar termos e relações. Em geral, desenvolve-se a ontologia de domínio a partir da hierarquia de uma ontologia de alto nível, e reaproveita-se termos de outras ontologias de domínio. Para desenvolvimento da HEMONTO, as ontologias selecionadas até o momento foram: (i) *Basic Formal Ontology* (BFO) [Grenon and Smith 2004], para a definição de classes genéricas; (ii) *Relation Ontology* (RO) [Smith et al. 2005], para definição de relações; e (iii) *Foundational Model of Anatomy* (FMA) [Rosse and Mejino 2003], para definição de parte das classes específicas de domínio. No restante desse artigo, apresentam-se as entidades em itálico e as relações em negrito.

2.2 Artefatos de representação: entidades e relações

Na construção da HEMONTO, os seguintes artefatos de representação têm sido utilizados: (i) *taxonomias*, correspondendo à relação formal ***is_a***; (ii) *partonomias*, correspondendo a relação formal **part_of**; (iii) *relações não-hierárquicas e outras*, tais como: **participates_in, has_agent, produces, has_quality**; e (iv) uso de uma *sintaxe semi-formal* para especificar as relações ontológicas, conforme recomendado pelo repositório *Open Biomedical Ontologies (OBO)*.

Tendo como base o padrão notacional da sintaxe adotada pela OBO, foram utilizadas 9 relações ontológicas da RO para ligar os termos da HEMONTO. Dessas 9 relações, sete são usadas nas estruturas de representação apresentadas neste artigo, e podem ser assim definidas formalmente [Smith et al. 2005]:

(1) *C* **is_a** *C1*: ∀c, ∀t, se c **instance_of** C at t então c **instance_of** C1 at t, tal que C, C1 suportam apenas entidades continuantes e c **instance_of** C at t é uma relação primitiva de instanciação, na qual a entidade continuante particular c instancia o universal C num dado tempo t.

(2) *P* **is_a** *P1*: ∀p, se p **instance_of** P então p **instance_of** P1, tal que P, P1 suportam apenas entidades ocorrentes e p **instance_of** P é uma relação primitiva de instanciação, na qual a entidade ocorrente particular p instancia o universal P.

(3) *C* **part_of** *C1*: ∀c, ∀t, se c **instance_of** C at t então há algum c1 tal que c1 **instance_of** C1 at t e c **part_of** c1 at t, onde c **part_of** c1 at t é uma relação primitiva entre dois particulares continuantes, na qual um é parte do outro no tempo t mencionado.

(4) *C* **participates_in** *P*: ∀c, ∀t, se c **instance_of** C at t então há algum p tal que p **instance_of** P e p **has_participant** c at t, onde **has_participant** c at t é uma relação primitiva entre um processo e um continuante num tempo t, tal que o continuante participa do processo de alguma maneira.

(5) *P* **produces** *C*: ∀p, se p **instance_of** P então há algum c e algum t; tal que se c **instance_of** C1 at t and p **produces** c at t, onde p **produces** c at t é uma relação entre o processo p, o continuante c e um tempo t, no qual p **produces** c se algum processo que **occurs_in** p **has_output** c.

(6) *P* **preceded_by** *P1*: ∀p, se p **instance_of** P então há algum p1 tal que p1 **instance_of** P1 and p **preceded_by** p1, onde p **preceded_by** p1 = ∀t, t1, se p **occurring_at** t and p1 **occurring_at** t1, então t1 **earlier** t, onde t **earlier** t1 é uma relação primitiva entre dois tempos tal que t ocorre antes de t1 e p **occurring_at** t = for some c, p **has_participant** c at t.

(7) *C* **has_quality** *Q*: é uma relação entre um continuante C e uma qualidade Q, a qual C **has_quality** Q se somente se: ∀c, ∀t, se c **instance_of** C at t então há algum c1 tal que: se c1 **instance_of** C1 and exists ∀q, ∀t, se q **instance_of** Q at t então há algum q1 tal que: se q1 **instance_of** Q1, tal que q **inheres_in** c at t.

(8) *Q* **is_quality_measured_as** *q*: é uma relação entre um universal continuante Q e um particular continuante q, sendo que ambos são qualidades e ∀q, ∀t, se q **instance_of** Q at t então há algum q1 tal que q1 **instance_of** Q1.

Sobre essas definições formais das relações apresentadas, cabe ressaltar que: (i) as variáveis c, c1, C e C1 representam entidades continuantes; (ii) p, p1, P e P1 entidades ocorrentes; (iii) q, q1 e Q representam qualidades; (iv) t, e t1 representam instâncias de tempo; e (v) variáveis minúsculas, tais como p e c, correspondem a particulares e variáveis maiúsculas, tais como P e C, correspondem a universais.

3. Resultados

A HEMONTO representa o conhecimento sobre hemocomponentes e hemoderivados, englobando elementos constituintes e os procedimentos empregados para sua obtenção. Em sua versão atual, a ontologia possui 53 termos, dos quais 44 são classes da ontologia e 9 são relações. Dentre as classes, trinta e três classes são específicas da ontologia, 9 classes foram importadas da BFO e 2 classes da FMA. As 9 relações são importadas da RO. O editor de ontologias Protege 4.2[1] foi utilizado para a construção da ontologia e possibilitou a sua implementação na linguagem *Ontology Web Language* (OWL). Para a construção dos diagramas da ontologia proposta foi utilizado o software Diagram Editor[2]. No restante dessa seção, descreve-se o conhecimento utilizado para criar a ontologia e apresentam-se diagramas.

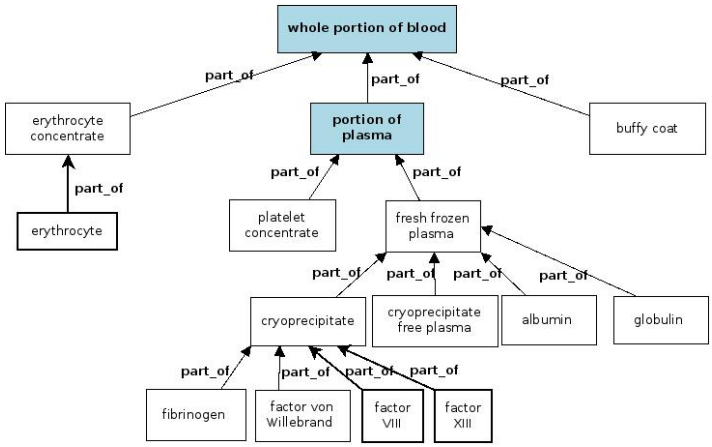


Figura 1: Partonomia dos componentes do sangue humano.

O *sangue (FMA:portion of blood)* é o líquido principal do corpo humano, formado de *plasma (FMA:portion of plasma)* e células sanguíneas. Para obter

[1] Disponível em: http://protege.stanford.edu/. Acesso em: 03 de Setembro de 2013.
[2] Disponível em: https://projects.gnome.org/dia/. Acesso em: 03 de Setembro de 2013.

hemocomponentes e hemoderivados do sangue é necessário submeter uma unidade de *sangue total (FMA:whole portion of blood)* a processos de centrifugação e congelamento. De acordo com o FMA, o sangue total é um tipo de uma porção de sangue, cujos seus componentes não foram separados. Esses tipos de hemocomponentes e hemoderivados estão representados em uma partonomia (veja figura 1).

Na figura 1, os retângulos sombreados representam as entidades *FMA:whole portion of blood* e *FMA:portion of plasm*. O *whole portion of blood* é submetido à centrifugação e separado inicialmente em três produtos: (i) *erythrocyte*, que armazenado em certas condições de temperatura gera o hemocomponente *erythrocyte concentrate*; (ii) *FMA:portion of plasma*, que corresponde ao *plasma* em sua situação natural (bruta), ainda rico em plaquetas (termo sinônimo: *platelet rich plasma)*; e (iii) *buffy coat*, porção do sangue formada por leucócitos e plaquetas. Após um novo processo de centrifugação em alta rotação, a *portion of plasma* é separado em: (i) hemocomponente *platelet concentrate* e (ii) o *fresh frozen plasma*, com baixa porcentagem de plaquetas. O *fresh frozen plasm* pode ser submetido a um processo de extração de um de seus componentes – o *cryoprecipitate* – dando origem a dois outros hemocomponentes: (i) o próprio *cryoprecipitate* e (ii) o *cryoprecipitate-free plasm*. Do *fresh frozen plasma* ainda é possível extrair dois hemoderivados (i – *albumin* e ii – *globulin*), a partir do fracionamento desse plasma em processos industriais. Por fim, o *cryoprecipitate* contém uma série de glicoproteínas de alto peso molecular (*fibrinogen, factor Von Willdebrand, factor VIII* e *factor XIII*) que cumprem o papel de fatores de coagulação na hemoterapia. A partir dessas proteínas é possível gerar o hemoderivado *clotting factors concentrate*.

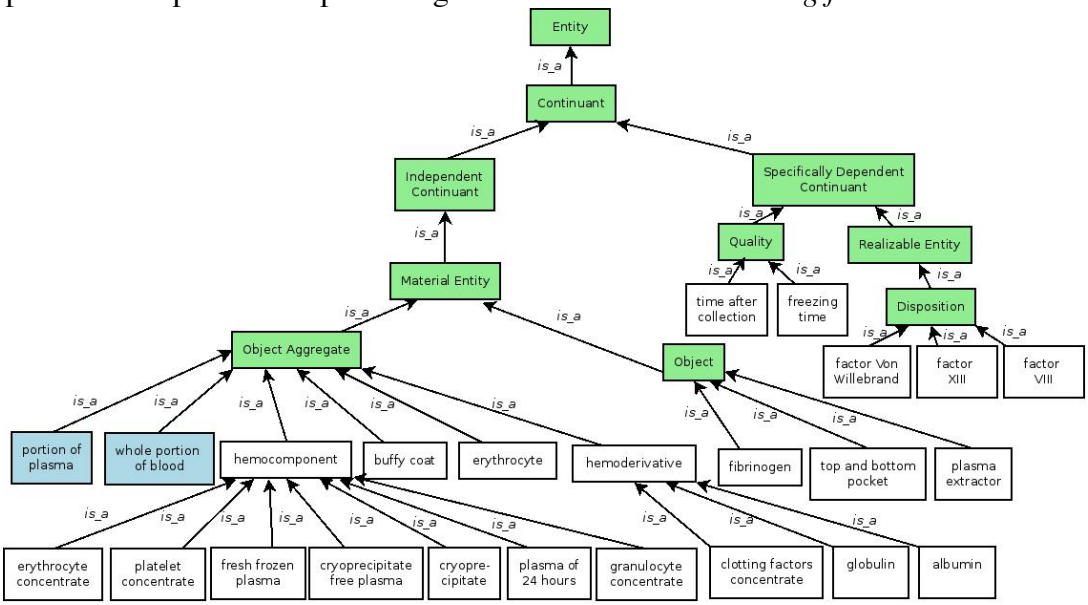


Figura 2: Taxonomia das entidades continuantes da HEMONTO.

De acordo com a estrutura taxonômica da ontologia de alto nível adotada, a BFO, tem-se dois grandes grupos de entidades: (i) *continuantes*, entidades que persistem ao longo do tempo mantendo sua identidade e que não possuem partes temporais; e (ii) *ocorrentes*, entidades que se revelam, se manifestam, ou se desenvolvem ao longo do tempo e possuem partes temporais [Spear 2006]. Seguindo tal estrutura, foram criadas duas taxonomias das entidades incluídas na HEMONTO. A

taxonomia da figura 2 representa o conjunto de *entidades continuantes* da HEMONTO, na qual as entidades importadas da BFO e do FMA estão sombreadas.

A entidade *FMA:whole portion of blood is_a BFO:object aggregate*, da mesma forma que seus componentes iniciais (*FMA:plasma, buffy coat* and *erythrocyte*). São também classificados como *BFO:object aggregate* os *hemocomponents* e os *hemoderivatives* e seus tipos específicos: (i) *erythrocyte concentrate, platelet concentrate; fresh frozen plasma; cryoprecipitate free plasm, cryoprecipitate, plasma of 24 hours* e *granulocyte concentrate*; e (ii) *clotting factors concentrate, globulin* e *albumin*. As entidades *plasma extractor*, *top and bottom pocket* e a proteína *fibrinogen* são classificadas como *BFO:object*, e os fatores de coagulação *factor Von Willebrand, factor VIII* and *factor XIII*, são classificados como *BFO:disposition*. Por fim, as entidades *time after collection* and *freezing time* foram classificadas como *BFO:quality*.

O outro grande grupo de entidades da HEMONTO corresponde aos *ocorrentes*, as quais representam os processos para obtenção dos hemocomponentes: *process of freezing; process of defrosting; process of centrifugation, process of centrifugation at high rotation*; *process of collection*; *extraction of buffy coat; remotion of plasm* and *remotion of cryoprecipitate*. Todas essas entidades foram classificadas como *BFO:process*. Por limitações de espaço, os processos não são representados na figura.

O plasma é um dos mais importantes componentes do sangue humano e, a partir dele, são gerados quatro hemocomponentes do sangue, conforme mostrado na figura 3, a seguir. Os retângulos da figura 3 representam classes da ontologia e as elipses representam propriedades destas classes.

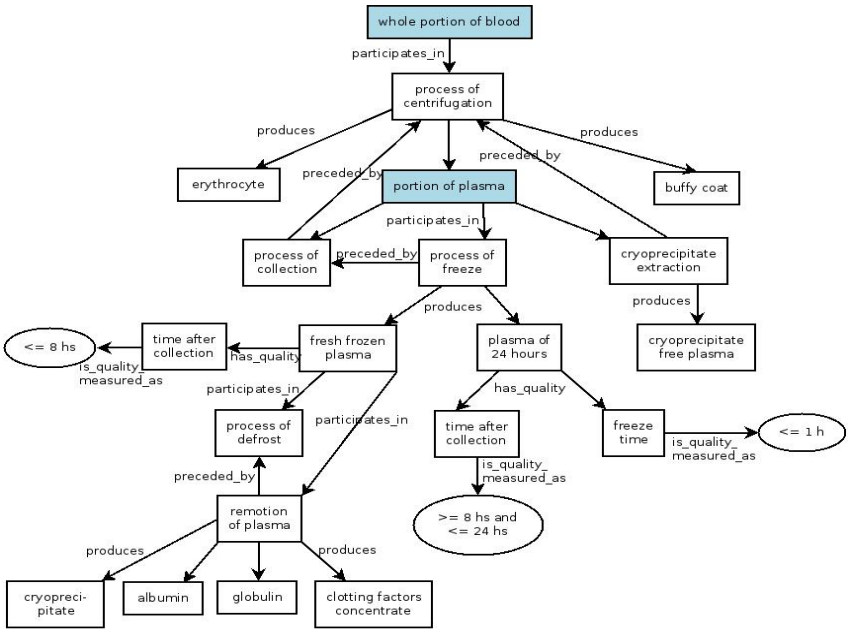


Figura 3: Processos de obtenção dos componentes plasmáticos

Para obtenção dos hemocomponentes *fresh frozen plasma* e *plasma of 24 hours* (lado esquerdo, figura 3), o procedimento inicial é o *process of centrifugation* do *whole portion of blood* para a separação de *erythrocyte, buffy coat* e *portion of plasma*. O passo seguinte consiste no *process of collection* da porção de plasma obtida. O tempo decorrido após a coleta do plasma, denominado de *time after collection*, é um parâmetro importante no processo, pois determina qual hemocomponente será obtido: (a) se o

tempo é no máximo 8 horas obtém-se o *fresh frozen plasm* e (b) quando o tempo pós-coleta está entre 8 horas e 24 horas obtém-se o *plasma of 24 hours*. Para que esses hemocomponentes sejam gerados é necessário que após o processo de coleta, o plasma seja submetido ao *process of freezing*. No caso do *plasma of 24 hours*, o tempo de congelamento deve ser no máximo de 1 hora, ou seja, *plasma of 24 hours* **has_quality** *freeze time* **is_quality_measured_as** <= 1 h. Ainda nesse diagrama, representa-se a obtenção do *cryoprecipitate free plasma* e do próprio *cryoprecipitate*. Para obter o *cryoprecipitate free plasma* (lado direito, figura 3), a etapa inicial corresponde novamente ao *process of centrifugation* do *whole portion of blood*, em seguida, tem-se o processo de *cryoprecipitate extraction* do plasma. Após essa extração obtém-se o *cryoprecipitate free plasma*. Já o *cryoprecipitate* é obtido a partir do *fresh frozen plasm* (lado esquerdo, figura 3) à temperatura de 1º C a 6º C. Esse plasma é submetido ao *process of defrosting* e, em seguida, o plasma sobrenadante é removido (*remotion of plasma*), deixando-se na bolsa coletora apenas a proteína precipitada e 10-15 ml deste plasma. Esses produtos formam o *cryoprecipitate*. No processo de *remotion of plasma*, o plasma removido **produces** *albumin*, *globulin* e *clotting factors concentrate*.

4. Considerações finais

O presente artigo apresentou um estudo de caso no domínio do sangue humano através da descrição do conteúdo parcial de uma ontologia sobre hemocomponentes e hemoderivados do sangue. A etapa seguinte desta pesquisa consiste na validação do conteúdo da ontologia, por parte de profissionais especialistas e a incorporação de novos termos à ontologia. A fim de possibilitar a validação do conteúdo da HEMONTO por parte dos especialistas, será criada uma interface de busca aos termos da ontologia, disponibilizada na web.

Referências

Almeida, M. B.; Proietti, A. B.; Smith, B. and Ai, J. (2011). "The Blood Ontology: an ontology in the domain of hematology". In: *ICBO 2011*; Buffalo, USA.

Brasil. Ministério da Saúde(2008)."Guia para o uso de hemocomponentes".Brasília, DF.

Grenon, P.; Smith, B. (2004). "SNAP and SPAN: Towards Dynamic Spatial". *Spatial Cognition & Computation*, v.4, n.1, p. 69-104.

Guarino, N. (1998). "Formal Ontology and Information Systems". In: *FOIS'98*; november 20, 2007; Trento, Italy. Edited by Guarino, N. IOS Press; pp. 3-15.

Rosse, C.; Mejino, J. L. V.(2003)."A reference ontology for biomedical informatics: the Foundational Model of Anatomy". *Journal of Biomedical Informatics*, 36:478-500.

Smith, B.; Ceusters, W.; Klagges, B.; Köhler, J.; Kumar, A.; Mungall, C.; Rector, A. L.; Rosse, C. (2005). "Relations in biomedical ontologies". *Genome Biology* 6, R46.

Smith, B.; Ceusters, W. (2010). "Ontological realism: A methodology for coordinated evolution of scientific ontologies". *Applied Ontology* 5, p. 139–188. IO Press.

Spear, A. D. (2006). "Ontology for the twenty first century: an introduction with recommendations". Saarbrücken, Germany.

Wand, Y.; Weber, R. (2004). "Reflection: Ontology in Information Systems". *Journal of Database Management* 15 (2), iii-vi.

Arquitetura de um Sistema de Recomendação Baseado em Ontologia para Anúncios de Carros

Fábio A. P. de Paiva[1], José A. F. Costa[2], Cláudio R. M. Silva[3], Ricardo S. França[4]

[1, 2, 4] Departamento de Eng. Elétrica – Universidade Federal do RN (UFRN)
Natal – RN – Brasil

[3] Departamento de Eng. de Comunicações – Universidade Federal do RN (UFRN)
Natal – RN – Brasil

fabiopaiva@yahoo.com,
{jafcosta, claudio.rmsilva, ricardoluizsf}@gmail.com

Abstract. *Recommender systems have emerged as one interesting approach to tackle the problem of information overload, however most of they have a problem. They fail when there are no identical keywords for an exact match of a search. In order to overcome this limitation, recently several proposals for systems have been presented. Some of them have proposed the integration of ontologies to improve the recommendation process. This paper presents an architecture for an ontology-based system and implements a prototype which demonstrates how it can be used to inside a portal to sell cars.*

Resumo. *Os sistemas de recomendação surgiram como uma abordagem interessante para resolver o problema da sobrecarga de informação. Entretanto a maioria deles falha quando não há palavras-chave idênticas para uma correspondência exata em uma pesquisa. Para minimizar essa limitação, recentemente várias novas propostas têm sido apresentadas. Algumas delas têm procurado explorar os benefícios das ontologias no processo de recomendação. Este trabalho apresenta uma arquitetura de um sistema baseado em ontologias e utiliza uma implementação de protótipo para demonstrar como ela pode ser usada em um portal de vendas de veículos.*

1. Introdução

Diariamente os usuários da Internet e suas aplicações criam cerca de 2,5 quintilhões de bytes de dados. Algumas estimativas calculam que 90% dos dados de hoje foram criados nos dois últimos anos [Zikopoulos *et al.* 2012]. Nesse contexto, um dos principais desafios de um usuário web é identificar informações que atendam às suas preferências e é por isso que os serviços personalizados de recomendação tornaram-se cada vez mais necessários [Kang e Choi 2011] e amplamente utilizados em várias áreas.

Os sistemas de recomendação surgiram como uma abordagem para resolver o problema da sobrecarga de informação. Eles são considerados aplicações especiais que fornecem sugestões personalizadas sobre produtos (ou serviços) que podem ser interessantes aos usuários. Os sistemas tradicionais sugerem itens (*e.g.*, uma música, um filme ou um livro) usando técnicas de mineração de texto [Gruber 1993]. No entanto, esses sistemas falham quando não há palavras-chave idênticas, mesmo existindo uma relação semântica entre elas [Kang e Choi 2011]. Para minimizar esse problema, nos últimos anos, vários estudos propuseram o uso de ontologias [Gao *et al.* 2008], [Zhen *et*

al. 2010], [Kang e Choi 2011], [Ge *et al.* 2012] como uma maneira de aumentar o desempenho dos sistemas de recomendação.

Neste trabalho, é proposta uma arquitetura de um sistema de recomendação baseado em ontologia o qual fornece informações personalizadas por meio dos relacionamentos entre os interesses do usuário e os anúncios de carros disponíveis na web. Na seção 2, são apresentadas a arquitetura e a forma de cálculo do grau de interesse; na seção seguinte, a técnica e o mecanismo de recomendação utilizados são descritos e, na última seção, as considerações finais são apresentadas.

2. Arquitetura Proposta

A Figura 1 apresenta a arquitetura hierárquica proposta de um sistema de recomendação baseado em ontologias para auxiliar usuários na compra de carros usados.

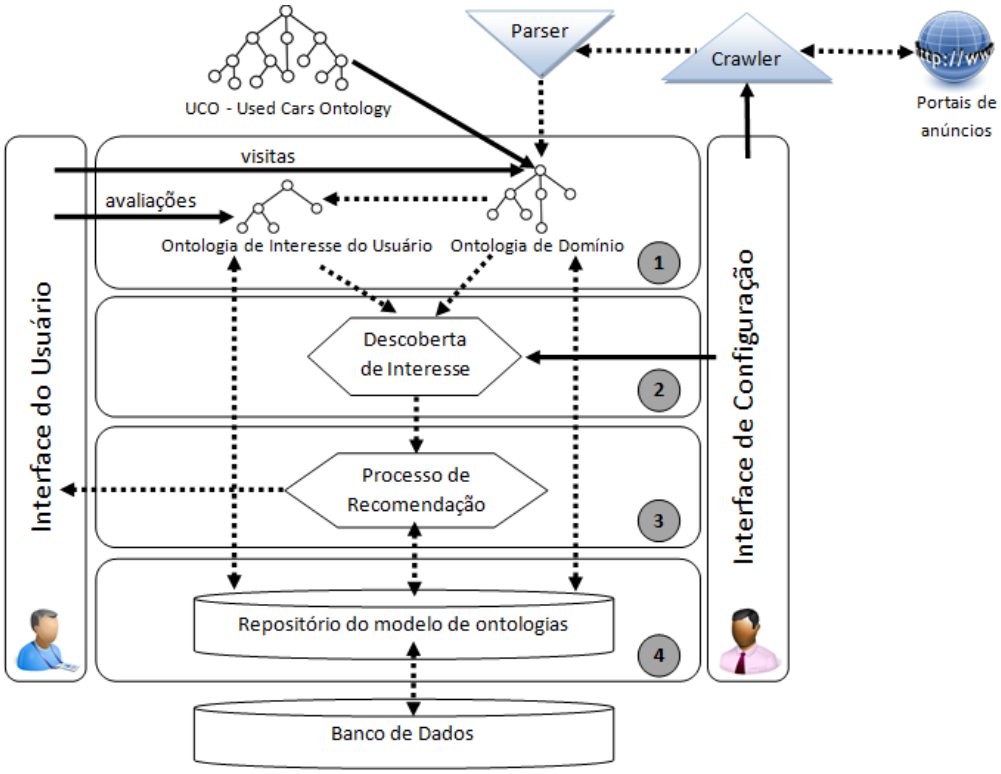


Figura 1. Arquitetura proposta do sistema

A arquitetura é baseada em quatro camadas, conforme a descrição abaixo:

1. Camada de Contexto — formada pelas ontologias de domínio (subseção 2.1) e de interesse do usuário (subseção 2.2);

2. Camada de Descoberta de Interesse — a partir das avaliações do usuário, é calculado o grau de interesse pelos conceitos da ontologia;

3. Camada de Recomendação — responsável por: a) calcular a similaridade entre os usuários, b) prever itens que serão interessantes para um determinado usuário e c) *rankear* a lista de anúncios de acordo com o interesse do usuário;

4. Camada de Ontologias — responsável pelo armazenamento das instâncias que representam os artefatos dos modelos de ontologia utilizados na arquitetura.

2.1. Ontologia de Domínio

Algumas vezes é interessante que apenas uma parte de uma ontologia seja reaproveitada. Este processo é chamado de modularização e consiste na extração de um subconjunto (também chamado de módulo) da ontologia original [Doran 2006]. Aqui, está sendo usado um módulo do modelo *Used Cars Ontology* (UCO) [MakoLab 2012].

Neste trabalho, um web *crawler* é encarregado de localizar anúncios web referentes a carros usados. O conteúdo descoberto nos anúncios é usado como entrada para construir uma lista de itens disponíveis para recomendação. Depois que o conteúdo dos anúncios é devidamente extraído por um componente com a função de *parser*, os anúncios são usados para popular a ontologia de domínio. A configuração do web *crawler* é definida pelo administrador do sistema na interface de configuração.

2.2. Ontologia de Interesse do Usuário

A ontologia que representa os interesses do usuário é um subconjunto da ontologia de domínio e, para construí-la, é realizado um mapeamento entre os interesses do usuário e os conceitos da ontologia de domínio (Figura 2). A fim de tornar as recomendações mais personalizadas, o usuário avalia os itens de acordo com as suas preferências.

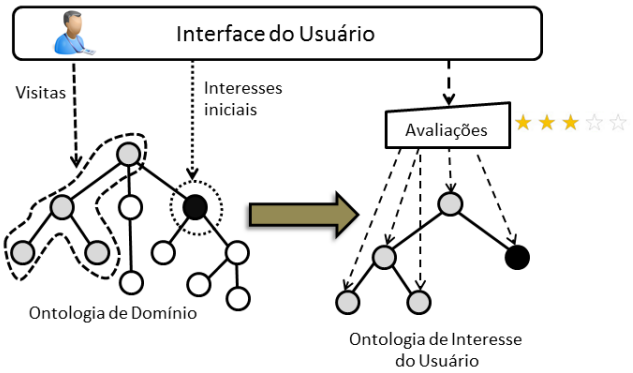


Figura 2. Processo de construção da Ontologia de Interesse do Usuário

É importante garantir que as características dos itens recomendados combinem com os interesses do usuário [Ge *et al.* 2012] a fim de garantir recomendações que atendam às suas necessidades. O interesse de cada usuário é representado por um modelo chamado de Modelo de Interesse do Usuário e formalmente pode ser definido como:

$$\Theta = (D, I, G, S), \quad \text{onde:}$$

- D = {login, sexo, estadoCivil, anoNascimento, numeroFilhos, renda} representa os dados pessoais do usuário;
- $I = \{I_1, I_2, ..., I_n\}$ é o conjunto de interesses (interesses iniciais + interesses descobertos através de interações) representados na ontologia do usuário;
- $G = \{G_1, G_2, ..., G_n\}$ é o grau de interesse do usuário por cada um dos conceitos de I e;
- $S = [S_{ij}]_{n \times n}$ é a matriz que representa a similaridade entre todos os usuários.

2.3. Grau de Interesse do Usuário

Depois de apresentadas as ontologias de domínio e de interesses, é calculado o grau de interesse do usuário por cada um dos conceitos de sua ontologia. Através de um formulário, realiza-se a avaliação dos itens {ótimo, bom, razoável, ruim, péssimo} de acordo com a pontuação {5, 4, 3, 2, 1}, respectivamente. O *feedback* explícito (avaliações) de um usuário em relação a um conceito C, representado por $\mathrm{Exp}(C_i)$, está no intervalo [-1, 1] e é calculado de acordo com a Equação 1:

$$\mathrm{Exp}(C_i) = \frac{\mathrm{pontuacao}\ (C_i) - 3}{2} \quad (1)$$

Em seguida, $\mathrm{Exp}(C_i)$ é normalizado para o intervalo [0, 1]. Já o *feedback* implícito (visitas), é baseado nas frequências de acesso e é calculado conforme Equação 2:

$$\mathrm{Imp}(C_i) = \frac{F(C_i)}{\sum_{j=1}^{n} F(C_j)} \quad (2)$$

onde F(Ci) é a frequência de acesso aos conceitos da ontologia de interesse. Por fim, o grau de interesse em relação a um conceito C, $G(C_i)$, é calculado pela Equação 3:

$$G(C_i) = \alpha * \mathrm{Exp}(C_i) + \beta * \mathrm{Imp}(C_i), \quad (3)$$

onde α e β são pesos que influenciam diretamente no cálculo do grau de interesse do usuário. A soma desses pesos é igual a 1 ($\alpha + \beta = 1$) e os seus valores são definidos pelo administrador do sistema.

3. Mecanismo de Recomendação

3.1. Filtragem Colaborativa

Após o cálculo do grau de interesse, os usuários são agrupados de acordo com suas similaridades e, para tal, são utilizados os Mapas Auto-Organizáveis e o algoritmo *K-Means*. O objetivo é que o sistema utilize a Filtragem Colaborativa para recomendar a um determinado usuário anúncios considerados interessantes baseado na opinião de outros usuários que apresentam perfis similares ao seu.

Os Mapas Auto-Organizáveis (*Self-Organizing Maps* ou simplesmente SOM) são algoritmos de redes neurais artificiais que se baseiam no aprendizado competitivo não-supervisionado, o que significa que o treinamento é inteiramente orientado pelos dados e que leva em consideração apenas os padrões de entrada [Kohonen 1997]. Cada neurônio *i* do mapa é representado por um vetor de peso *p*-dimensional $m_i = [m_{i1}, m_{i2}, ..., m_{ip}]^T$, onde *p* é igual à dimensão do vetor de entrada [Costa e Netto 2001].

A Matriz-U é um método de visualização usada, normalmente, com os mapas SOM para análise de agrupamentos. Ela se baseia na distância do espaço de entrada entre um vetor de peso e os seus vizinhos no mapa [Yamaguchi e Ichimura 2011]. Já a Matriz-U* [Ultsch 2003] leva em consideração a densidade da informação a fim de aprimorar os resultados da Matriz-U convencional.

3.2. Resultados Preliminares

O vetor de entrada do algoritmo SOM é formado pelo conjunto de grau de interesses do usuário, $G(C_i) = \{C_1, C_2, ..., C_n\}$, e por alguns de seus dados pessoais, representados numericamente. Os dados utilizados nos experimentos foram obtidos de forma

simulada. No entanto, a fim de aproximá-los da realidade, eles foram gerados a partir de uma distribuição normal e, em seguida, normalizados na faixa de valores [0, 1].

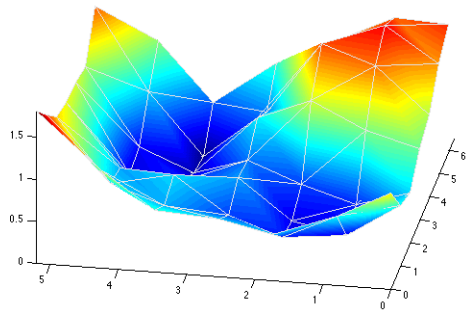


Figura 3. Matriz-U* construída a partir dos dados de entrada do SOM

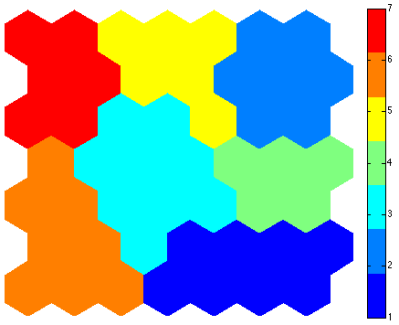


Figura 4. Mapa SOM representando 7 grupos de usuários similares

O conjunto de dados foi analisado por um mapa SOM, empregado na construção da Matriz-U* (Figura 3). O algoritmo *K-Means* foi utilizado para segmentar a U*. E, para determinar o número adequado de agrupamentos, usou-se o CDbw (*Composed Density between and within clusters*) [Halkidi e Vazirgiannis 2008], um índice que avalia a compacidade e a separação de grupos definidos por um algoritmo de agrupamento. Ao fim da execução, são obtidos sete grupos que representam os perfis de usuários similares (Figura 4) e a matriz de similaridade correspondente, $[S_{ij}]_{n \times n}$.

3.3. Processo de Recomendação

A listagem de recomendação é gerada a partir de regras definidas pelo administrador. A performance do sistema é determinada por essas regras que atuam como parâmetros de configuração. Alguns exemplos das regras usadas são a) o *threshold* que determina o interesse (ou não) do usuário por um anúncio, b) a atribuição de valores aos pesos α e β (ver seção 2.3), c) os parâmetros de inicialização do algoritmo SOM e outros.

Quando um usuário *n* é similar a um usuário *u*, pode-se dizer que *n* é um vizinho de *u*. Depois de realizado o agrupamento de usuários, o próximo passo é prever os itens *i* que ainda não foram visualizados pelo usuário *u*, mas que já foram avaliados anteriormente pelos seus vizinhos *n*, conforme Equação 4 [Schafer *et al.* 2007]:

$$previsao(u,i) = \overline{r_u} + \frac{\sum_{n \in vizinhos(u)} simUsuario(u,n) * (r_{ni} - \overline{r_n})}{\sum_{n \in vizinhos(u)} simUsuario(u,n)} \quad (4)$$

onde $\overline{r_u}$ e $\overline{r_n}$ são, respectivamente, as avaliações médias dos usuários *u* e *n*, enquanto r_{ni} é a avaliação do usuário *n* sobre o item *i*. Finalmente, os itens recomendados são ordenados baseado no valor de $G(C_i)$.

4. Conclusões

Neste trabalho, é apresentada uma arquitetura hierárquica de um sistema baseado em ontologias para recomendação de anúncios de carros usados. A arquitetura utiliza duas ontologias: uma de domínio e outra para representar os interesses do usuário. A fim de aplicar a técnica de Filtragem Colaborativa, um mapa SOM é empregado para agrupar usuários com características e interesses similares. Enquanto os sistemas de

recomendação tradicionais utilizam palavras-chave para representar os interesses do usuário, este trabalho utiliza conceitos de ontologia. Dessa forma, o modelo de interesses será mais adequado à realidade do usuário e consequentemente o mecanismo de recomendação sugerirá um maior número de anúncios que atendam às reais necessidades do usuário. O trabalho ainda está em fase de desenvolvimento e a continuidade na implementação de outras funcionalidades é o foco de trabalhos futuros.

Referências

Costa, J. A. F., Netto, M. L. A. (2001). Clustering of complex shaped data sets via Kohonen maps and mathematical morphology. In Proceedings of the SPIE, Data Mining and Knowledge Discovery. B. Dasarathy (Ed.), vol. 4384, pp. 16-27.

Doran, P. (2006). Ontology reuse via ontology modularisation. *In Proceedings of Knowledge Web PhD Symposium*, pp. 1-6.

Ge, J., Chen, Z., Peng, J. e Li, T. (2012). An ontology-based method for personalized recommendation. *In 11th International Conference on Cognitive Informatics & Cognitive Computing* (ICCI*CC), pp.522-526.

Gao, Q., Yan, J. e Liu, M. (2008). A Semantic Approach to Recommendation System based on User Ontology and Spreading Activation Model". *In International Conference on Network and Parallel Computing*, pp. 488-492.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Journal Knowledge Acquisition*, pp. 199-220.

Halkidi, M. e Vazirgiannis, M. (2008). A density-based cluster validity approach using multi representatives. *Pattern Recognition Letters*, vol. 29, pp. 773-786.

Kang, J. e Choi, J. (2011). An ontology-based recommendation system using long-term and short-term preferences. *In International Conference on Information Science and Applications*, pp. 1-8.

Kohonen, T. (1997). Self-Organizing Maps. 2nd. Ed., Berlim: Springer, Verlag.

Makolab [site] (2012). Used Cars Ontology Metadata. Disponível em: http://ontologies.makolab.com/uco/ns.html. Acessado em: 15 jun. 2013.

Schafer, J. Ben, Frankowski, D., Herlocker, J. e Sen, S. (2007). The adaptive web, P. Brusilovsky, A. Kobsa e W. Nejdl, Springer-Verlag, Alemanha, p. 291-324.

Yamaguchi, T. e Ichimura, T. (2011). Visualization using multi-layered U-Matrix in growing Tree-Structured self-organizing feature map. Systems, Man and Cybernetics (SMC), IEEE International Conference, p. 3580-3585.

Ultsch, Alfred. (2003). U*-Matrix: a Tool to visualize Cluster in high dimensional Data. Technical Report No. 36, Dept. of Mathematics and Computer Science, University of Marburg, Germany.

Zikopoulos, P., Eaton, C., Deutsch, T., Deroos, D. e Lapis, G. (2012). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. New York: McGraw-Hill, pp. 176.

Zhen, L., Huang, G. Q. e Jiang, Z. (2010). An inner-enterprise knowledge recommender system. *In Expert Systems with Applications*, vol. 37, pp. 1703-1712.

Ontologia de Contexto e Qualidade de Contexto

Débora Cabral Nazário[1,2], Mário Antônio Ribeiro Dantas[1], José Leomar Todesco[1]

[1]Engenharia e Gestão do Conhecimento (EGC)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

[2]Departamento de Ciência da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)
Joinville – SC – Brasil

debora.nazario@gmail.com, mario.dantas@ufsc.br, tite@egc.ufsc.br

Abstract. *This paper approaches Context Knowledge Representation and Quality of Context (QoC) through the development of an ontology in this domain. It is thereby intended to improve knowledge sharing between humans and machines, and allow knowledge reuse between applications. A QoC assessment can contribute in order to enable context-aware applications to detect anomalies in sensors, generate alerts, discard data with low QoC, choose an appropriate provider, activate backup sensors, and other actions. It is also expected that the use of ontology will help identify potential QoC problems.*

Resumo. *Este artigo aborda a representação de conhecimento de Contexto e Qualidade de Contexto (QoC) através do desenvolvimento de uma ontologia neste domínio. Desta forma, pretende-se melhorar o compartilhamento de conhecimento entre humanos e máquinas e permitir o reúso de conhecimento entre aplicações. A avaliação de QoC pode contribuir para que aplicações sensíveis ao contexto sejam capazes de detectar anomalias nos sensores, gerar alertas, descartar dados com QoC insuficientes, escolher um provedor adequado, ativar sensores backup, entre outras ações. Espera-se ainda, que a utilização de ontologia auxilie na identificação de possíveis problemas de QoC.*

1. Introdução

A computação ubíqua é um paradigma que está cada vez mais fazendo parte das atividades diárias das pessoas, através do uso de dispositivos móveis ou portáteis. Este tipo de computação possui forte ligação com as características do mundo físico e dos perfis de seus usuários. Estas informações são chamadas de contextos e representam o elemento de entrada para a computação ciente ou sensível ao contexto.

Desta forma, um sistema pode utilizar estas informações de contexto relevantes e consequentemente prover serviços mais otimizados e personalizados, aumentando a satisfação dos usuários. Também é possível minimizar o consumo de recursos como energia, processamento e comunicação através da utilização do contexto, disponibilizando serviços mais precisos e dinâmicos [Loureiro et al. 2009].

Em ambientes ubíquos, um dos muitos fatores importantes é a sensibilidade de contexto. Mas as informações de contexto podem não ser confiáveis ou úteis, apresentando um problema de qualidade da informação de contexto. Sendo assim, um ponto importante na sensibilidade de contexto é que a informação de contexto seja confiável [Kim and Lee 2006]. Ou seja, é necessário que a qualidade das informações de contexto ou possíveis problemas sejam conhecidos.

A qualidade das informações de contexto utilizadas na adaptação de serviços tem um impacto significativo sobre as experiências dos usuários com serviços sensíveis ao contexto, que pode ser positivo ou negativo, dependendo da Qualidade de Contexto (QoC). Desta forma, a QoC pode auxiliar o usuário a estimar o comportamento de um serviço sensível ao contexto, também pode servir como um indicador para a seleção de um provedor de contexto mais adequado.

Neste trabalho de pesquisa é abordada a representação de conhecimento de contexto e QoC através do desenvolvimento de uma ontologia neste domínio. O artigo está organizado da seguinte forma: A seção 2 aborda os conceitos relacionados a Contexto e QoC, a seção 3 descreve a ontologia proposta, a seção 4 apresenta alguns resultados iniciais, os resultados esperados e trabalhos futuros, finalmente têm-se as referências utilizadas.

2. Contexto e Qualidade de Contexto

O contexto é qualquer informação que possa ser utilizada para caracterizar a situação de entidades como: pessoa, lugar ou objeto, que sejam consideradas relevantes para interação entre um usuário e uma aplicação [Dey 2000].

Para Chen and Kotz (2000), o contexto apresenta quatro dimensões. O contexto computacional lida com os aspectos técnicos, relacionados com capacidades e recursos computacionais; o contexto físico é acessível por meio de sensores e recursos como: localização, condição de tráfego, velocidade, temperatura, iluminação entre outros; contexto de tempo capta informações de tempo, como de um dia, semana, mês, estação do ano, ano, etc.; o contexto do usuário está relacionado à dimensão social do usuário, como seu perfil, pessoas nas proximidades, situação social, preferências.

A Qualidade de Contexto (QoC) é qualquer informação que descreve a qualidade da informação que é usada como informação de contexto [Buchholz et al. 2003]. QoC não está exigindo informação de contexto perfeita, com a maior precisão possível e atualidade, mas é necessária uma estimativa correta da qualidade da informação [Bellavista et al. 2012].

Informações de contexto de alta qualidade desempenham um papel fundamental na adaptação de um sistema que apresente mudanças repentinas. No entanto, a diversidade das fontes de informação de contexto e as características dos dispositivos de computação impactam fortemente na qualidade de informações de contexto em ambientes de computação pervasiva [Manzoor et al. 2008].

Na etapa inicial deste estudo foi realizada uma detalhada revisão da literatura sobre Qualidade de Contexto, gerando uma taxonomia das publicações que abordam QoC [Nazário et al. 2012a].

Na sequência, outro trabalho abordou a Representação de Conhecimento de Contexto e Qualidade de Contexto, onde foram identificados modelos que utilizam notação gráfica, marcação XML (*Extensible Markup Language*), UML (*Unified*

Modelling Language) e ontologias e OWL (*Ontology Web Language*) [Nazário et al. 2012b].

Destas abordagens para representação de contexto destaca-se a ontologia, por permitir o compartilhamento de conhecimento entre humanos e agentes de *software*, além de possibilitar a reutilização de conhecimento entre aplicações e a sua utilização por motores de inferência.

Dando continuidade ao trabalho, uma ontologia de contexto considerando QoC é proposta, conforme descrito a seguir.

3. Ontologia de Contexto e QoC

Com base nos estudos realizados, optou-se pelo uso de ontologia, pois alguns trabalhos desta área já utilizam esta abordagem, e no que se referem à QoC, os trabalhos são iniciais, pouco aprofundados, como em [Tang et al. 2007] [Toninelli and Corradi 2009], [Filho et al. 2010]. Sendo assim, se percebe oportunidade para avanço nas pesquisas, com possíveis contribuições, principalmente aprimorando a ontologia com os benefícios de um modelo ontológico.

O processo de construção de ontologia empregado nessa pesquisa foi baseado na Metodologia 101 [Noy and Deborah L. McGuinness 2001]. Para o desenvolvimento da ontologia foi utilizada a plataforma Protégé-OWL.

Com relação ao domínio e escopo da ontologia, é considerado o contexto, a QoC e focado em um cenário de ambiente ubíquo assistido, onde é monitorada a saúde do usuário.

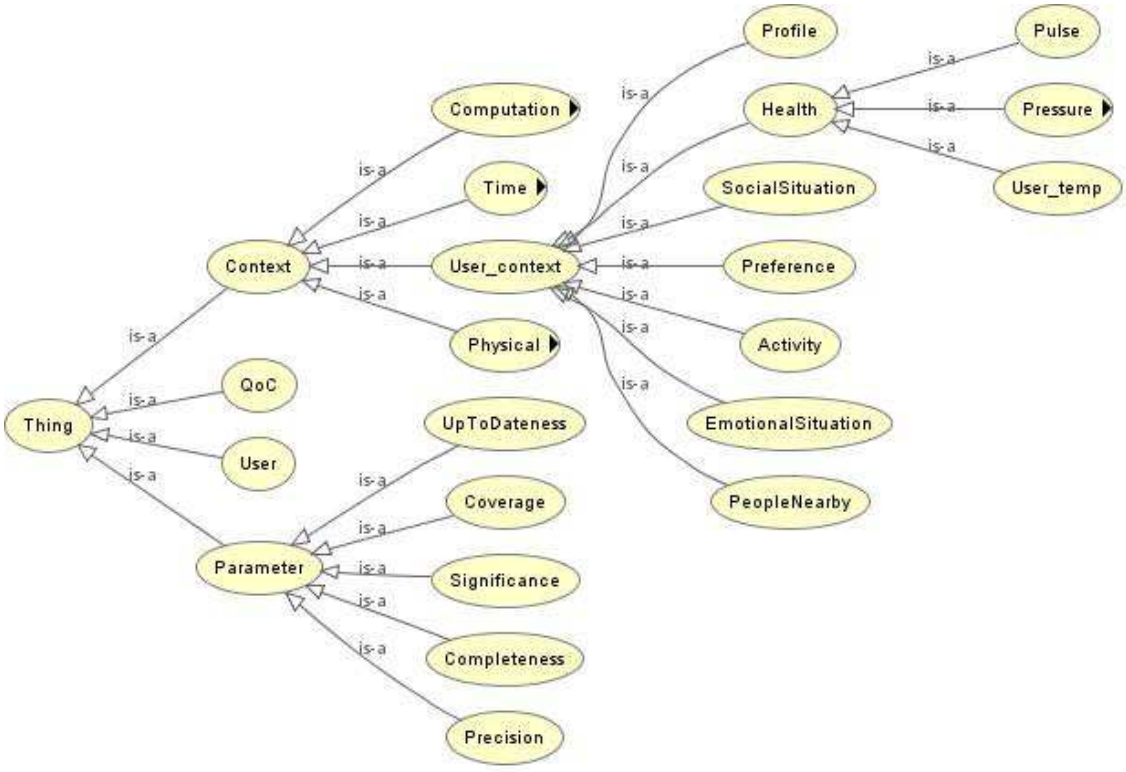


Figura 1. Hierarquia de Classes

No desenvolvimento desta ontologia utilizou-se a classificação de contexto de Chen and Kotz (2000) para definição das classes, além de outras ontologias de contexto da literatura como: [Kim and Choi 2006], [Escobedo 2008] e ontologias de QoC: [Toninelli and Corradi 2009], [Filho et al. 2010].

Na Figura 1 está representada a hierarquia de algumas classes criadas, como: *Context*, QoC, *User*, *Parameter* e suas respectivas subclasses. Algumas subclasses ocultadas da figura são: *Computation* (*Device*, *Network*, *Resource*), *Physical* (*Humidity*, *Location*, *Luminosity*, *Noise*, *Pollution*, *Temperature*, *Traffic*), *Time* (*Day*, *Hour*, *Minute*, *Month*, *Second*, *Year*).

Os parâmetros de QoC foram selecionados da literatura de acordo com a sua relevância e forma de quantificação. Em síntese, pode-se dizer que as informações de QoC do sensor representam o quanto a informação fornecida é:

- Atual, através do parâmetro *Up-to-dateness*;
- Válida, parâmetro *Coverage*;
- Significante, parâmetro *Significance*;
- Completa, parâmetro *Completeness*;
- Precisa, parâmetro *Precision*;

A Figura 2 apresenta o relacionamento entre as Classes da ontologia. As relações criadas entre os indivíduos das classes são: ***hasUserContext***: classe *User* e classe *User_context*; ***hasTime***: classe *User_context* e classe *Time*; ***hasQoC***: classe *Context* e classe *QoC*; ***hasParameter***: classe *QoC* e classe *Parameter*; ***hasSignificance***: classe *Parameter* e classe *Significance*.

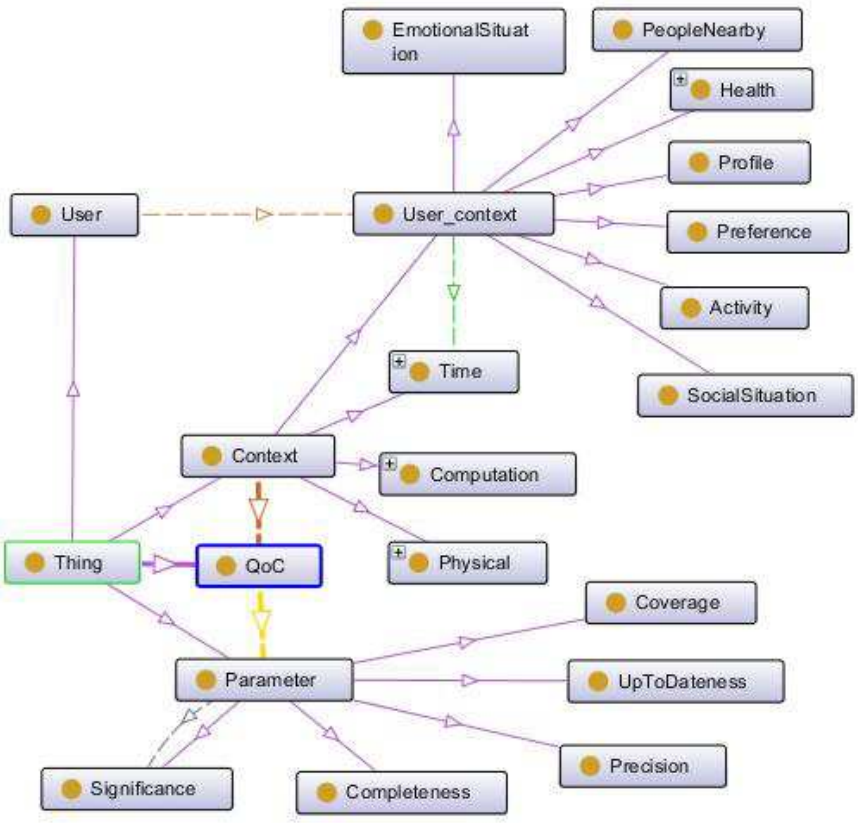


Figura 2. Relacionamento entre as Classes

4. Resultados e Considerações

Algumas instâncias foram criadas na ontologia, baseadas em uma simulação de um ambiente ubíquo assistido, onde foram consideradas nesta etapa as informações: nome do usuário, sua temperatura corporal, o tempo, valor da QoC e de seus parâmetros. Alguns testes foram realizados com a ferramenta SPARQL *query*, como o exemplo do Quadro 1, que seleciona instâncias com valor de QoC inferior a 0.5.

Quadro 1. Exemplo de Busca por Instâncias

```
SELECT ?name ?user_temp ?valueQoC ?P ?value
      WHERE { ?user ont:hasUserContext ?temp.
      ?user ont:name ?name.
      ?temp ont:f_value ?user_temp.
      ?temp ont:hasQoC ?QoC.
      ?QoC ont:f_value ?valueQoC.
      ?QoC ont:hasParameter ?P.
      ?P ont:f_value ?value
      FILTER (?valueQoC < 0.5) }
```

Na Figura 3, tem-se o resultado de uma busca onde são selecionadas instâncias com o parâmetro *Significance* igual a 1, o que indica um sinal de alerta a ser investigado. No primeiro caso a QoC é alta (0.99), indicando boa QoC, a alerta está no valor da temperatura, acima de 38 graus, indicando febre. Já no segundo caso, a QoC está baixa (0.66), indicando QoC não adequada ou insuficiente. Percebe-se um valor de temperatura não esperado (12.0), este é o motivo da alerta, o que sugere problema no sensor de temperatura. Nestes casos, os valores dos parâmetros de QoC podem ajudar a confirmar esta suspeita, como por exemplo através da precisão do equipamento.

name	user_temp	
"Debora"@	"38.1"^^<http://www.w3.org/2001/XMLSchema#decimal>	"0.99"^^<http://w
"Debora"@	"12.0"^^<http://www.w3.org/2001/XMLSchema#decimal>	"0.66"^^<http://w

Figura 3. Exemplo de Resultado de busca

Como resultados práticos espera-se que através da ontologia de QoC, uma aplicação sensível ao contexto possa: detectar anomalias ou inconsistências nos sensores, gerar alertas, ativar sensores *backup*, descartar dados com QoC insuficientes, escolher provedor adequado, entre outras ações.

Como trabalhos futuros pretende-se adicionar outras informações de contexto do usuário, assim como contextos de ambiente e dispositivos móveis, avaliar outros parâmetros de QoC, além de desenvolver a ontologia utilizada explorando mais os benefícios de um modelo ontológico. Espera-se que a ontologia aprimorada possa ser utilizada como modelo de dados em um sistema sensível ao contexto.

Agradecimento. Esta pesquisa é apoiada pelo *Programa do Fundo de Apoio à Manutenção e ao Desenvolvimento da Educação Superior* – FUMDES.

Referências

Bellavista, P., Corradi, A., Fanelli, M. and Foschini, L. (2012). A Survey of Context Data Distribution for Mobile Ubiquitous Systems. *ACM Computing Surveys*, v. 44, n. 4, p. 1–45.

Buchholz, T., Küpper, A. and Schiffers, M. (2003). Quality of Context : What It Is And Why We Need It. In *10th International Workshop of the HP OpenView University Association(HPOVUA)*.

Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. Technical Report.

Dey, A. K. (2000). Providing Architectural Support for Building Context-Aware Applications. Georgia Institute of Technology.

Escobedo, E. P. P. (2008). Modelagem de contexto utilizando ontologias. Universidade de São Paulo.

Filho, J. B., Miron, A. D., Satoh, I., Gensel, J. and Martin, H. (2010). Modeling and Measuring Quality of Context Information in Pervasive Environments. In *24th IEEE International Conference on Advanced Information Networking and Applications*.

Kim, E. and Choi, J. (2006). An ontology-based context model in a smart home. *Computational Science and Its Applications-ICCSA: LNCS*, v. 3983, p. 11–20.

Kim, Y. and Lee, K. (nov 2006). A Quality Measurement Method of Context Information in Ubiquitous Environments. In *International Conference on Hybrid Information Technology*. IEEE.

Loureiro, A. A. F., Augusto, R., Oliveira, Rabelo, et al. (2009). Computação Ubíqua Ciente de Contexto: Desafios e Tendências. In *27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.

Manzoor, A., Truong, H. and Dustdar, S. (2008). On the Evaluation of Quality of Context. In *3rd European Conference on Smart Sensing and Context*.

Nazário, D. C., Dantas, M. A. R. and Todesco, J. L. (2012a). Taxonomia das publicações sobre Qualidade de Contexto. *Sustainable Business International Journal*, v. 20, p. 1–28.

Nazário, D. C., Dantas, M. A. R. and Todesco, J. L. (2012b). Representação de Conhecimento de Contexto e Qualidade de Contexto. In *Jornada Iberoamericana de Ingeniería del Software e Ingeniería del Conocimiento*.

Noy, N. F. and Deborah L. McGuinness (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report.

Tang, S., Yang, J. and Wu, Z. (sep 2007). A Context Quality Model for Ubiquitous Applications. In *2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007)*. IEEE.

Toninelli, A. and Corradi, A. (2009). A Quality of Context-Aware Approach to Access Control in Pervasive Environments. *MobileWireless Middleware, Operating Systems, and Applications: LNCS*, v. 7, p. 236–251.

Conceptual Modeling of Formal and Material Relations Applied to Ontologies

Ricardo Ramos Linck, Guilherme Schievelbein and Mara Abel

Institute of Informatics – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91501-970 – Porto Alegre – RS – Brazil

{rrlinck,gschievelbein,marabel}@inf.ufrgs.br

Abstract. *Ontologies represent a shared conceptualization of a knowledge community. They are built from the description of the meaning of concepts, expressed through their attributes and their relationships. Relationships are used to describe how the concepts are structured in the world. This work reviews the literature on formal and material relations, especially on mereological and partonomic relations, and proposes an alternative for the conceptual modeling of such relations in a domain ontology. This alternative has been made available in the ontology building tool of the Obaitá Project.*

1. Introduction

This work falls in the area of Conceptual Modeling and Knowledge Engineering, focusing on the ontological foundations and conceptual modeling of relations applied to ontologies.

Ontology represents a shared conceptualization that includes concepts, its attributes and the relationships between the concepts. In addition to the subsumption relationships that build the taxonomies of concepts, other formal and material relations assist in structuring the domain and the conceptual definition. The main existing modeling tools, such as Protégé, WebODE and others, however, are still deficient in differentiating the various types of formal and material relationships in order to assign the possibilities of automated reasoning.

Obaitá Project is a tool for collaborative construction of visual domain ontologies based on foundational ontology. Continuing the development of the Obaitá ontology building tool, this work provides support to the ontological foundations of the relations, enforcing ontological consistency and providing visual component support into the ontology relations.

The main goals of this research project include providing:

- foundation ontological constructs to support the ontological choices of the kinds of relations through the semantic expressiveness of a foundational ontology, especially the formal (mereological and partonomic) and material relations;

- support to the inference of the ontological meta-type of the relations based on the meta-types of the respective related concepts;

- visual ontological constructs to represent the visual knowledge about relations among the ontology concepts, supporting imagistic domains;

- intuitive interface which, through the use of natural language, does not require users to have any prior knowledge of ontological representation formal languages.

Following, in Section 2, we present an overview of ontology and relations; in Section 3, we present an analysis on some of the main ontology building tools; in Section 4, we present our implemented solution; in Section 5, we present an example of use of the system; and finally, in Section 6, we present our conclusions about this work.

2. Ontology and Relations

In recent years, there has been a growing interest in the use of foundational ontologies for evaluating conceptual modeling languages, developing guidelines for their use and providing real-world semantics for their modeling constructs (Guizzardi and Wagner, 2010). One of the main foundational ontologies is UFO (Unified Foundational Ontology), which is divided into three incrementally layered compliance sets: UFO-A defines the core of UFO, as a comprehensive ontology of endurants; UFO-B defines, as an increment to UFO-A, terms related to perdurants; UFO-C defines, as an increment to UFO-A and UFO-B, terms related to the spheres of intentional and social entities (Guizzardi et al., 2007).

The importance of conceptual relationships is highlighted by (Bala and Aghila, 2011) when they state that relationships are fundamental to express semantics in ontology in order to associate concepts and associate instances. Relationships are defined according to their properties, like reflexivity, symmetry, transitivity. As argued in (Guarino, 2009) and (Grenon, 2003), relations can be divided into two broad categories, namely *formal* and *material* relations. Formal relations hold between two or more entities (*relata*) directly, without any further intervening individual. Figure 1 exemplifies a formal relation between alcohol and wine, where alcohol is part of wine.

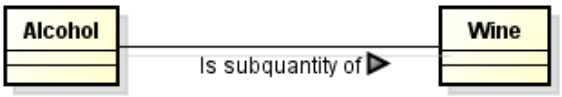


Figure 1. An example of formal relation.

Four sorts of conceptual formal part-whole relations are defined in (Guizzardi, 2005) with different semantics, based on the type of the related entities: component-of relates individuals that are functional complexes, subquantity-of relates individuals that are quantities, subcollection-of relates individuals that are collectives, and member-of relates individuals that are functional complexes or collectives (as part) and a collective (as a whole).

Parthood relationships are especially important for modeling visual knowledge, since the object recognition by cognitive systems that support vision is strongly based on composition and decomposition operations.

Unlike formal relations, material relations have material structure of their own and include examples such as working at; for a material relation of being treated in between Paul and a medical unit to exist, another entity must exist which mediates Paul and the medical unit. These entities are named *relators* (Guizzardi and Wagner, 2010). Figure 2 depicts an example of material relation between employee and company (*relata*), where, if an employee works for a company, another entity (*relator*), such as employment, must exist in order to mediate them.

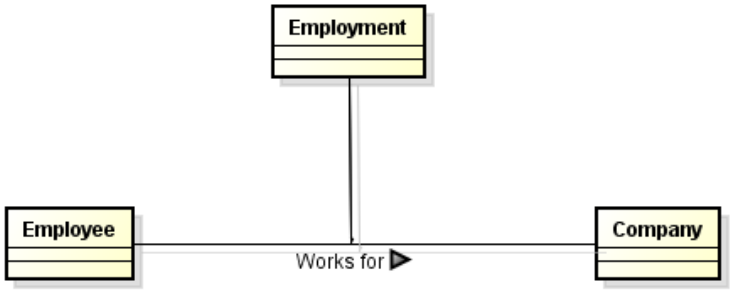


Figure 2. An example of material relation.

3. Relations in Ontology Building Tools

The main ontology building tools, such as Protégé and WebODE, among others, provide different support for specifying the ontology relations.

3.1. Protégé

Protégé 4 (Horridge et al., 2007) can compute subsumption relationships between classes, and detect inconsistent classes. It can be computed automatically by a reasoner. Binary relations, linking two individuals together, are represented by slots.

Properties describe binary relationships. There are two main types of properties: datatype properties and object properties. Datatype properties describe relationships between individuals and data values, and object properties describe relationships between individuals. Object properties may define some characteristics, such as functional, inverse functional, transitive, symmetric, asymmetric, reflexive, irreflexive.

Properties may present some restrictions, which fall into three main categories: quantifier, cardinality and hasValue restrictions. The quantifier restrictions effectively put constraints on the relations that the individual participates in. It does this by either specifying that at least one kind of relationship must exist (existential restrictions), or by specifying the only kinds of relationships that can exist (universal restrictions). The cardinality restrictions are the number of relationships that an individual may participate in for a given property. Cardinality restrictions may specify the minimum and the maximum cardinality restrictions. The hasValue restrictions describe the class of individuals that have at least one relationship to another specific individual.

3.2. WebODE

WebODE (Arpírez et al., 2001) allows the post-processing of the ontology, using the OntoClean methodology for identifying incorrect taxonomic (is-a) relations. WebODE works with both built-in relations and ad-hoc relations.

Built-in relations are predefined relations related to the representation of taxonomies of concepts and mereology relationships between concepts. They are divided into three groups: taxonomical relations between concepts, taxonomical relations between groups and concepts, and mereological relations between concepts. The taxonomical relations between concepts have two predefined relations: subclass-of and not-subclass-of. Single and multiple inheritance are allowed. The taxonomical relations between groups and concepts have two predefined relations: disjoint-subclass-partition and exhaustive-subclass-partition. The mereological relations between

concepts have two predefined relations: transitive-part-of and intransitive-part-of.

Ad-hoc relations are characterized by their name, the source and target concepts name, and its cardinality. WebODE allows just binary ad-hoc relations to be created between concepts. The creation of relations of higher arity must be made by reification.

3.3. Remarks about the tools

Analyzing the available tools, we noticed that most of them have both implementation and user interface oriented to formal languages of representation, like OWL, making it harder for users who do not have this expertise to use them properly. We also noticed that these tools do not include ontological foundations or visual domains.

The analyzed tools do not provide adequate support to the ontological choice problem: how to choose the best primitives to represent the needed relations. These issues may produce different specifications for the same conceptual model, or result in different interpretations of the same model by different users. Likewise, the construction of the relations in the user mind is strongly based in visual knowledge, but this topic is still incipient for the main ontology building tools. In the next section we describe the solution that has been implemented in order to achieve the goals of this research project.

4. Implemented Solution

This work supports the relation ontological foundations (according to UFO-A), enforces ontological consistency, provides inference, and provides visual component support.

Relations are specialized in formal or material relations, as seen in Figure 3. Material relations contain a *relator* and two *relata*. Formal relations contain two *relata*. The *relata* are existing concepts from the domain ontology, and the *relator* is a relational moment. Formal relations may be further specialized as part-whole relations (component-of, member-of, subcollection-of or subquantity-of), enforcing the following constraints: component-of, both *relata* are functional complexes (kind), they have to be irreflexive, asymmetric and nontransitive, and they have weak supplementation; member-of, the whole individual is a collective, while the part can be either a collective or a functional complex (kind), they have to be irreflexive, asymmetric and intransitive, and they have weak supplementation; subcollection-of, both *relata* are collectives, they have to be irreflexive, asymmetric and transitive, and they have weak supplementation; and subquantity-of, both *relata* are quantities, they have to be irreflexive, asymmetric and transitive, they have strong supplementation, and they have to be nonshareable.

When editing a concept relation, it is possible to choose its name, its type (classification by UFO-A), the target concept, the source and target cardinalities, the *relator* (for material relations) and its icon (visual component). The source concept is automatically selected as the concept that is being viewed in detail in the system.

In order to help users to define the relation type, the system guides them by asking questions, without requiring users to have any knowledge of ontological representation. For example, if he/she answers the question telling the system that the relation needs the existence of a mediating entity, then the relation type is “material”.

The system also has the ability to infer the relation type based on the meta-types of the respective related concepts. For example, if the meta-type of both related concepts is “quantity”, then the relation type is “subquantity-of”. Next, we present an

example of use through a real domain ontology, from the Sedimentary Stratigraphy area, in order to evaluate our research project proposed approach.

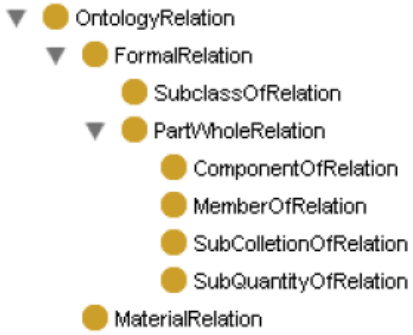


Figure 3. The meta-ontology relation structure.

5. Example of Use

In order to validate the system in a real environment, we brought an example of use in the Sedimentary Stratigraphy domain, an area of Geology responsible for studying the formation processes of sedimentary rocks. In (Lorenzatti et al., 2010), a domain ontology was built with the help of experts, serving as the basis for initiating the system. This domain has been chosen because it presents some important aspects for our focus: it is strongly based on visual knowledge; its structure is complex; and it has scientific and economic relevance, studying the generation and depositional conditions of important mineral deposits, such as coal and oil.

From this example of use, we intend to evaluate the approach proposed by our research project, considering the following parameters:

- Total of existing relations: “before” and “after” the activities performed by the geologists, classified by relation type (only “after” the activities performed by the geologists; the previous ontology relations were not based on ontological foundation).

- Total of changes performed on the relations: added relations, updated relations and removed relations, classified by relation type.

After these evaluations, then it will be possible to identify the contributions and resulting benefits from this research project approach regarding the ontological consistency of the created ontology concept relations. In the next section, we present our conclusions and some open possibilities for future improvement of this work.

6. Conclusions

The main contributions of this work include the definition of the ontological relations based on a set of metadata, providing specialized ontological constructs for creating the domain ontology relations and supporting the inference of the relation ontological meta-types. The ontology building environment is independent of the representation formal languages, providing intuitive interface so that users do not need any previous ontological representation knowledge in order to interact with the ontology. Some constructs allow the association of icons in order to obtain a higher domain understanding. This work takes in consideration the importance of the relation ontological foundations and the visual knowledge as supporting instruments.

As a result of our researches, our ontology building tool is constantly under improvement; we keep adding important features on its implementation, which many of them we do not find on most of the other tools. Thus, this specific research project has fundamental importance, continuing the evolution of an innovative tool for both academic and commercial purposes. An extensive evaluation on the modeling of the ontology relationships still has to be performed, as described in the previous section. Its benefits have already become explicit through the conceptual and intuitive approach added to the tool. The capabilities of the proposed metadata model will be assessed through a practical application by the construction of an ontology for the Sedimentary Stratigraphy domain from Geology.

This work can be considered as a step for future work in order to complement the ontological foundation of relations into the Obaitá ontology building tool, such as taking in consideration the taxonomic relations and the temporal relations.

Acknowledgement

This work is supported by the Brazilian Research Council (CNPq) and Petrobras PFRH.

References

Arpírez, J. C., Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2001). "WebODE: A Scalable Workbench for Ontological Engineering". In: Proceedings of the First International Conference on Knowledge Capture (K-CAP).

Bala, P. S. and Aghila, G. (2011). "Relationship Based Reasoning". In: International Journal of Computer Applications.

Grenon, P. (2003). "The Formal Ontology of Spatio-Temporal Reality and its Formalization". In: AAAI Technical Report Series, AAAI Spring Symposium on the Foundations and Applications of Spatio-Temporal Reasoning, Stanford University, Palo Alto, California, U.S.A.

Guarino, N. (2009). "The Ontological Level: Revisiting 30 Years of Knowledge Representation". Conceptual Modeling: Foundations and Applications, LNCS 5600, Springer-Verlag.

Guizzardi, G. (2005). "Ontological Foundations for Structural Conceptual Models". PhD Thesis, University of Twente, The Netherlands.

Guizzardi, G. and Wagner, G. (2010). "Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages". Theory and Applications of Ontology: Computer Applications, p.175-196, Springer.

Guizzardi, R. S. S., Guizzardi, G., Perini, A. and Mylopoulos, J. (2007). "Towards an Ontological Account of Agent-Oriented Goals". Software Engineering for Multi-Agent Systems V, Springer.

Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R. and Wroe, C. (2007). "A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools". Edition 1.1, University of Manchester.

Lorenzatti, A., Abel, M., Fiorini, S. R., Bernardes, A. K. and Scherer, C. M. S. (2010). "Ontological Primitives for Visual Knowledge". In: A.C. da Rocha Costa, R.M. Vicari, F. Tonidandel (Eds.): SBIA 2010, LNAI 6404, p.1-10, Springer-Verlag.

Integração de Padrões para Transferência de Informações Digitais no Fluxo de Trabalho de Modelagem de Reservatórios Baseada em Ontologias

Ricardo Werlang[1], **Mara Abel**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

{rwerlang,marabel}@inf.ufrgs.br

Abstract. *In the workflow of petroleum reservoir characterization, experts seek an integrated vision of the data issued from the same oil field, generated by the application of different techniques and represented in different standards and formats. Also, the same geological objects analyzed by different professionals assume distinct semantic representations and complementary in supporting decision-making. In this work, we aim to delimit the most used data and its formats in the construction of structural models and also to propose a semantic-based integrated approach using ontologies to capture the real meaning of the information and not just the specific technology used to represent it.*

Resumo. *No fluxo de trabalho de caracterização de reservatórios de petróleo, especialistas buscam uma visão integrada dos dados do mesmo campo petrolífero, originados pela aplicação de diferentes técnicas e representados em padrões e formatos diversos. Além disso, os mesmos objetos geológicos analisados por diferentes profissionais assumem significados semânticos e representações distintas e complementares no suporte à tomada de decisão. Nesse trabalho, pretende-se delimitar os dados e os formatos mais utilizados na construção de modelos estruturais e, também, propor uma abordagem de integração semântica utilizando ontologias que capture o significado real da informação e não apenas a tecnologia específica utilizada para representá-la.*

1. Introdução

O processo de modelagem de reservatórios de petróleo envolve uma complexa sucessão de atividades que dependem do tipo de modelo que é construído e das opções que são levadas em consideração para construí-lo. Estas atividades podem ser classificadas de acordo com o objetivo de estudo do modelo de reservatório em construção, que pode estar relacionado, segundo [Perrin and Rainaud 2013], com a geometria, na construção do modelo estrutural; com as características da rocha, na construção do modelo estratigráfico; e com os fluídos de um reservatório de petróleo, no modelo de reservatório.

Grande parte destas atividades, contudo, envolvem o uso de uma grande quantidade de dados, que são gerados diariamente por técnicas distintas e que necessitam ser analisados e interpretados pelos diferentes profissionais envolvidos na construção de modelos de reservatórios. No entanto, para usufruir do valor da informação contida nesses dados em diferentes formatos, os profissionais necessitam de acesso imediato às informações, de uma visão integrada das informações e de um completo gerenciamento do conhecimento já adquirido sobre as informações disponíveis [Soma et al. 2008]. Necessita-se,

portanto, de uma abordagem que permita a integração real dos objetos modelados e não apenas dos formatos de dados. Além disso, a abordagem de integração deve descrever tanto objetos quanto propriedades em uma linguagem uniforme, permitindo a todos os envolvidos o acesso às informações em qualquer etapa do processo de modelagem. Necessita-se, neste caso, de uma abordagem de integração semântica dos dados.

O uso de ontologias para modelar os diversos elementos do domínio, como rocha e petróleo, e de bases de conhecimento para armazenar as informações das instâncias desses elementos, como uma determina porção de rocha ou de petróleo, vem sendo proposto para resolver os problemas de integração de dados enfrentados nas operações de modelagem de reservatórios de petróleo. Além disso, nas últimas décadas, diversos padrões abertos ou proprietários foram propostos e utilizados para transferência de informações digitais entre as diferentes etapas e atividades envolvidas no fluxo de trabalho de modelagem de reservatórios. Esses padrões foram criados para resolver o problema de dados criados em diferentes sistemas com formatos proprietários. A integração dos formatos evidenciou o problema de que objetos geológicos podem assumir diferentes significados em etapas distintas da exploração de petróleo. Embora sejam muitas vezes referenciados pelo mesmo vocabulário, são definidos por atributos distintos, que buscam atender o papel daquele objeto em uma etapa particular da exploração.

Nesse trabalho, serão definidas as atividades que envolvem a interpretação de diferentes formatos de dados no processo de construção de um modelo estrutural e, também, proposta uma abordagem de integração semântica, utilizando ontologias, que resolva os principais problemas enfrentados no processo de modelagem.

O restante do texto está organizado da seguinte maneira: a Seção 2 apresenta os padrões para troca de informações mais utilizados na indústria de petróleo; a Seção 3 contém uma visão geral da cadeia de exploração de petróleo, destacando as atividades da modelagem estrutural; a Seção 4 comporta o uso de ontologias e a abordagem de integração proposta; e a Seção 5 conclui o trabalho, expondo os trabalhos futuros.

2. Padrões para Trocas de Informações Digitais

A criação de uma plataforma de trabalho comum sempre foi uma das maiores preocupações dos fornecedores de softwares para a indústria do petróleo. No entanto, do ponto de vista do usuário, ainda é muito difícil transferir dados de uma plataforma para outra. O principal motivo é o histórico complicado do mercado de geomodelagem, caracterizado por extensões ou reconstruções de produtos, aquisições de softwares e fusões de empresas. Perrin *et al.*, em [Perrin and Rainaud 2013], apresenta a evolução histórica das ferramentas para modelagem, que começaram a surgir em meados dos anos 80, com o objetivo de representar as superfícies geológicas e as propriedades petrofísicas das rochas.

No início de 1990, as descrições de reservatórios só poderiam ser trocadas através da escrita e leitura de arquivos de dados proprietários, que transportavam informações limitadas. Na tentativa de facilitar a comunicação entre os diversos softwares utilizados no fluxo de trabalho para construção de modelos da Terra, diversos estudos foram feitos e algumas tentativas de padronização foram propostas. O objetivo da criação de padrões, que são definidos pelos metadados que dão significado e contexto às informações representadas, é permitir que a semântica dos dados seja revelada.

Entre os padrões mais utilizados na indústria, destaca-se o *Log ASCII Standard*

(LAS), que foi proposto em 1990, pela *Canadian Well Logging Society*[1]. O LAS é um padrão utilizado para facilitar e simplificar a troca de informações digitais de dados de *log* de poços. Por ter sido proposto no formato ASCII, que possibilita a importação e exportação para qualquer plataforma, o LAS teve ampla aceitação e utilização na indústria. Contudo, padrões mais recentes, que utilizam tecnologias que promovem a interoperabilidade de uma maneira mais natural, como o XML (*eXtensible Markup Language*), vêm sendo propostos nas últimas décadas e estão sendo cada vez mais utilizados. A Energistics[2] tem sido a líder na exploração do XML dentro da indústria do petróleo, através da proposta de diversos padrões: o WITSML, como um padrão para transferência de informações de perfuração de poços; o PRODML, para transferência de dados de operações de produção; e o RESQML, para transferência de dados contendo descrições de modelos.

3. Atividades de Modelagem de Reservatórios para Exploração de Petróleo

Exploração de petróleo é uma atividade na qual a aquisição, a distribuição e o uso do conhecimento dos especialistas são críticos para a tomada de decisão. Modelos geológicos são ferramentas chaves para a identificação e caracterização de potenciais reservatórios de hidrocarbonetos. Modelos geológicos são representações, 3D ou 4D, de dados e interpretações relacionadas com recursos do subsolo e são desenvolvidos por diferentes profissionais, como geólogos, geofísicos e engenheiros, que são responsáveis pela evolução de um potencial reservatório de petróleo, através de várias etapas de modelagem conhecida como fluxo de trabalho de modelagem de reservatórios [Mastella 2010].

O fluxo de trabalho de modelagem de reservatórios começa com a definição da área de interesse a ser modelada, conhecida como prospecto. É realizada, então, a aquisição dos dados sísmicos e de perfuração de poços, assim como dos documentos e mapas da geologia regional. Com estes dados, é realizada a construção do modelo estrutural, que é essencial para determinar a localização de armadilhas de hidrocarbonetos e, consequentemente, para a identificação de possíveis campos petrolíferos e para uma possível avaliação do volume de óleo disponível. A Figura 1 ilustra as principais atividades realizadas na construção do modelo estrutural, como a interpretação de log de poços, as interpretações geológica e sísmica, o encaixe das marcações de poços e a modelagem de superfície. A descrição detalhada das atividades pode ser encontrada em [Perrin and Rainaud 2013]. Pelo fato de todas estas atividades envolverem uma grande quantidade de dados, enfatizamos os principais formatos de dados utilizados. Entre estes dados, alguns tendem a seguir um determinado padrão, como o LAS e o WITSML, que são os mais utilizados. Outros dados, contudo, não seguem praticamente nenhuma estrutura, como os tipos DAT, SEG-Y, PLO e documentos, planilhas, anotações e mapas geológicos.

Após a construção do modelo estrutural, são construídas as malhas estratigráficas dentro de cada bloco geológico definido no modelo estrutural. Estas malhas são divididas por diversas células, que devem carregar as propriedades petrofísicas das rochas. Para isso, são utilizados dados de estudos de laboratório. Estas propriedades são propagadas para todo o volume utilizando simulações geo-estatísticas. O resultado deste processo é o modelo estratigráfico. Este modelo, juntamente com o resultado do processo de aprimoramento da malha estratigráfica, resulta no modelo de reservatório, utilizado para simular a quantidade de óleo acumulado no subsolo.

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

[2]http://www.energistics.org/

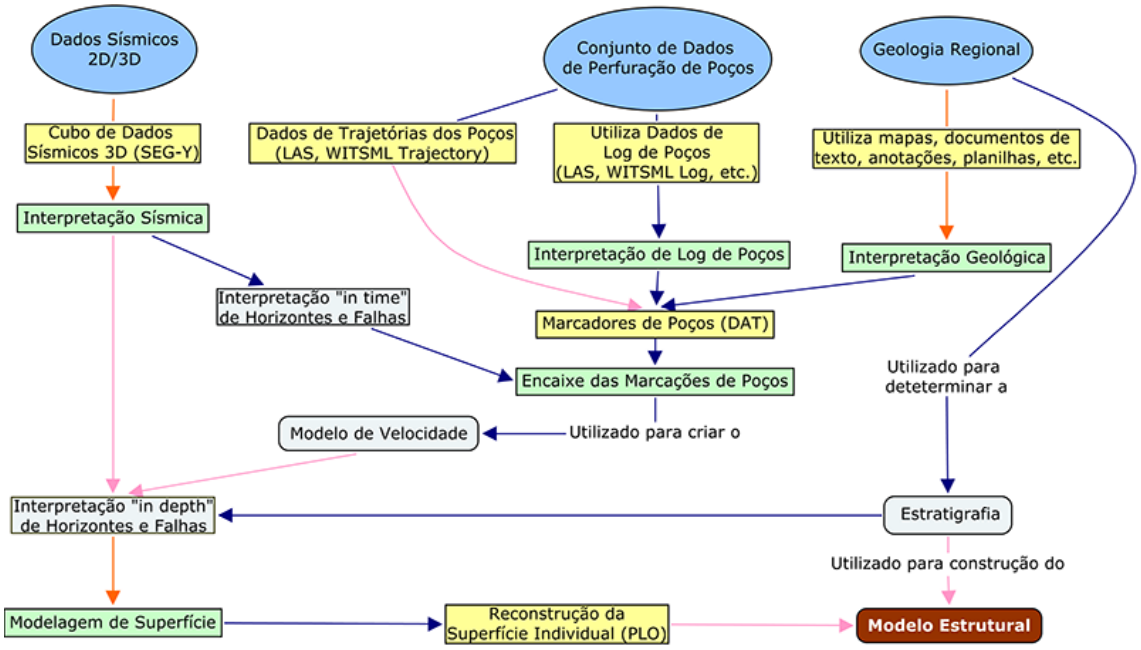


Figura 1. Construção do Modelo Estrutural [Perrin and Rainaud 2013]

4. Integração Baseada em Ontologias

Diversas entidades, companhias de petróleo e consórcios de Geociências vêm emitindo diversas codificações e formalizações do conhecimento geológico durante os últimos anos. Entre os levantamentos geológicos, destacam-se: o modelo GeoSciML, que é uma formalização baseada na normativa GML (*Geography Markup Language*), para a representação de características geográficas e geométricas; o NADM (*North American Geologic Map Data Model*), projetado como uma ontologia para desenvolvimento interoperável de banco de dados geológicos centrados em mapas; e o projeto GEON (*Geosciences Network*)[3], para a integração de mapas geológicos, cujos arquivos de origem contêm informações com esquemas e vocabulários diferentes. Entre projetos de companhias de petróleo, mesmo não englobando áreas das Geociências, merece destaque a ontologia proposta no projeto IPP (*Integrated Information Platform*) para formalização de terminologias usadas na etapa de produção de petróleo, baseada no padrão ISO 15926.

Além disso, muitas ontologias foram propostas para domínios específicos, tanto ontologias de domínio quanto ontologias de nível superior. Entre as ontologias de domínio, destacam-se: a ontologia proposta em [Abel et al. 2004], para descrição petrográfica de rochas de reservatório; a ontologia proposta em [Lorenzatti et al. 2009], para modelagem de estruturas sedimentares e atributos textuais de rochas; e a ontologia proposta em [Perrin et al. 2005], que descreve os principais conceitos utilizados na modelagem estrutural. Entre as ontologias de nível superior, destacam-se: a ontologia proposta pelo projeto SWEET (*Semantic Web for Earth and Environmental Terminology*), desenvolvida na NASA, que fornece milhares de termos sobre todo o sistema da Terra [Raskin and Pan 2005]; e a ontologia proposta em [Mastella 2010], a *Basic Geology*, que descreve e interconecta as entidades geológicas consideradas na modelagem de reservatórios. A *Basic Geology* utiliza a *GeoLocation*, uma ontologia com termos geográficos;

[3]http://www.geongrid.org/

ontologias para as disciplinas de *Paleogeography*, *Hydrogeology* e *Lithology*; e ontologias para definição e mapeamento de eras geológicas, *Geological Time* e *Geological Dating*.

Nesse trabalho, propomos a reutilização dessas ontologias em destaque, objetivando a construção de um modelo conceitual único que formalize tanto os conceitos geológicos, que estão envolvidos no processo de modelagem da Terra, quanto os conceitos gerais do domínio. Utilizamos, para isso, o uso da abordagem de ontologia única para integração de informações [Wache et al. 2001]. Contudo, no trabalho apresentando em [Wache et al. 2001], o autor assume que todas as fontes de dados são bancos de dados. Nosso problema, no entanto, é mais complexo, uma vez que as informações que desejamos integrar estão representadas por diferentes formatos de arquivos e armazenadas em diferentes fontes de informações, que devem ser mantidas nos locais e formatos originais [Mastella 2010]. Além disso, em grandes indústrias de petróleo, os profissionais envolvidos na modelagem necessitam de uma maneira eficiente para encontrar os dados desejados, o que não corresponde à realidade atual. A forma de busca ideal é através de consultas relacionadas aos significados reais dos dados, isto é, pela semântica dos dados.

Identificou-se, portanto, a necessidade de resolver dois problemas: (i) localizar os dados desejados e (ii) mapear as informações desses dados com o modelo conceitual proposto. A fim de resolver o primeiro problema, propomos o uso de metadados, que são definidos como dados sobre dados. Desse modo, para cada objeto de dados (arquivos LAS, WITSML, DAT, PLO, Documentos Geológicos, etc.), são definidos dois tipos de metadados: (i) metadados de informações de acesso, que definem como os objetos de dados podem ser acessados, isto é, o local, o nome e o tipo (extensão) do arquivo; (ii) metadados de proveniência, que descrevem como os objetos de dados foram criados, incluindo o autor, a dada de criação e a data de última modificação.

Uma vez com os metadados de informações de acesso e proveniência, os objetos poderão ser processados por um analisador sintático, específico para cada padrão. Considerando como gramática a sintaxe do padrão utilizado no arquivo, que declara precisamente quais são os possíveis elementos a serem descritos, este analisador sintático poderá auxiliar no mapeamento entre os objetos de dados e o modelo conceitual proposto. Nesse processo, as informações dos arquivos deverão ser transformadas em instâncias ontológicas, que serão armazenadas em uma base de conhecimento. Dessa maneira, os usuários poderão realizar buscas nessa base de conhecimento através de consultas relacionadas aos dados geológicos representados pelos objetos de dados.

5. Conclusão

Apresentamos, neste artigo, a necessidade de uma abordagem de integração dos objetos de dados utilizados no fluxo de trabalho de modelagem de reservatórios. Esta abordagem, contudo, deve levar em consideração o significado real dos dados e não apenas a tecnologia utilizada para representá-los. Para isso, identificamos os principais objetos de dados e padrões utilizados no processo e realizamos um levantamento das principais ontologias disponíveis para esse domínio. Propomos a reutilização dessas ontologias para a construção de um modelo conceitual único, que é uma ferramenta chave para atender as necessidades dos profissionais envolvidos no processo de modelagem: encontrar, de maneira eficiente, os dados desejados, isto é, através da realização de consultas pelos significados reais dos dados. Identificamos, então, os dois principais problemas enfrentados

nas indústrias de petróleo e propomos soluções: (i) utilizar metadados para auxiliar na localização dos arquivos desejados e (ii) extrair as informações dos dados, descritos por padrões, para auxiliar os profissionais no mapeamento das informações geológicas, que estão representadas nos objetos de dados, com o modelo conceitual proposto.

No estado atual de desenvolvimento do trabalho, focamos no processo de construção do modelo estrutural. Em trabalhos futuros, pretendemos estender esse estudo para todo processo de modelagem de reservatórios. Para isso, iremos analisar a construção dos modelos estratigráfico e de reservatório, a fim de definir os principais padrões utilizados e uma forma de mapeá-los ao modelo conceitual. A validação do modelo de integração proposto será realizada através do desenvolvimento de um protótipo que permita o cadastramento de novos objetos de dados e, posteriormente, a recuperação destes objetos de dados através de consultas que relacionem os objetos geológicos que eles representam.

6. Agradecimentos

Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES, pelo Programa de Excelência Acadêmica - PROEX, assim como pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq.

Referências

Abel, M., Silva, L. A., De Ros, L. F., Mastella, L. S., Campbell, J. A., and Novello, T. (2004). Petrographer: managing petrographic data and knowledge using an intelligent database application. *Expert Systems with Applications*, 26(1):9–18.

Lorenzatti, A., Abel, M., Nunes, B. R., and Scherer, C. M. (2009). Ontology for imagistic domains: Combining textual and pictorial primitives. In *Advances in Conceptual Modeling-Challenging Perspectives*, pages 169–178. Springer.

Mastella, L. S. (2010). *Semantic Exploitation of Engineering Models: Application to Petroleum Reservoir Models*. PhD thesis, Ecole Nationale Superieure des Mines De Paris.

Perrin, M. and Rainaud, J.-F. (2013). *Shared Earth Modeling: Knowledge Driven Solutions for Building and Managing Subsurface 3D Geological Models*. Editions Technip.

Perrin, M., Zhu, B., Rainaud, J.-F., and Schneider, S. (2005). Knowledge-driven applications for geological modeling. *Journal of Petroleum Science and Engineering*.

Raskin, R. G. and Pan, M. J. (2005). Knowledge representation in the semantic web for earth and environmental knowledge representation in the semantic web for earth and environmental terminology (sweet). *Computers & Geosciences*.

Soma, R., Bakshi, A., Prasanna, V., DaSie, W., and Bourgeois, B. (2008). Semantic web technologies for smart oil field applications. In *Intelligent Energy Conference and Exhibition*.

Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S. (2001). Ontology-based integration of information-a survey of existing approaches. In *IJCAI-01 Workshop: Ontologies and Information Sharing*.

Rede de Pesquisadores Brasileiros em Ontologia: Uma Análise de Rede Social

Andréa S. Bordin[1], Alexandre Leopoldo Gonçalves[1]

[1]Departamento de Engenharia e Gestão do Conhecimento – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

andreabord@gmail.com,a.l.goncalves@ufsc.br

Abstract. *The research interest in the topic ontology has grown considerably in recent decades. Ontologies are searched or used primarily in the areas of Humanities, Exact and Health. This article aims to analyze the collaboration network based on co-authorship of Brazilian researchers on this topic. We retrieved 1179 articles in journals and national and international conferences from the Scopus database. The co-authorship network was created and analyzed using metrics social network analysis. The results show the existence of a fragmented network with many individual components, where the major component of the area presents authors with higher computing centralities.*

Resumo. *O interesse de pesquisa pelo tópico ontologia vem crescendo consideravelmente nas últimas décadas. Ontologias são pesquisadas ou utilizadas principalmente nas áreas de Ciências Humanas, Exatas e Saúde. Esse artigo objetiva analisar a rede de colaboração baseada em coautoria dos pesquisadores brasileiros nesse tópico. Foram recuperados 1179 artigos em periódicos e conferências nacionais e internacionais a partir da base de dados Scopus. A rede de coautoria foi criada e analisada através de métricas de análise de rede social. Os resultados mostram a existência de uma rede fragmentada com muitos componentes isolados, onde o maior componente apresenta autores da área de computação com as maiores centralidades.*

1. Introdução

Ontologia vem se tornando um tópico de interesse de pesquisadores em todo o mundo. Inicialmente abordado por pesquisadores da área de Ciências Humanas pelas suas raízes na Filosofia, posteriormente se transformou em objeto de pesquisa da área de Computação.

Uma pesquisa recente pelo termo "ontolog*" na base de dados Scopus revelou a existência em torno de 56.000 documentos contra 2 documentos em 1980. A mesma pesquisa, para documentos onde pelo menos um autor é brasileiro encontrou 1179 documentos, sendo o primeiro deles publicado no ano de 1995.

A partir desse ano, o número de artigos com a participação de pelo menos um autor brasileiro teve uma ascendência constante e atingiu seu pico em 2010, com 198 documentos, o que denota o interesse e a importância desse tópico de pesquisa para a ciência no Brasil.

Esse artigo analisa a colaboração cientifica entre os pesquisadores brasileiros em ontologia. Estudar a colaboração cientifica de um campo de conhecimento permite que sejam descobertos os focos e fluxos de transferência de conhecimentos e habilidades do grupo que colabora (KATZ; MARTIN, 1997). O objetivo deste trabalho é descobrir os autores que mais colaboram e as sub-redes de colaboração. O indicador utilizado é o de coautoria dos documentos coletados na base Scopus. Com isso, criou-se a rede de coautoria dos pesquisadores e o método de análise de rede social foi aplicado. Os resultados obtidos buscam fornecer o perfil da área de pesquisa em ontologia no Brasil e com isso ajudam novos e atuais pesquisadores a se posicionar nesse panorama.

A seguir será apresentado o referencial da área de análise de rede social, os procedimentos metodológicos, os resultados, a discussão e a conclusão do trabalho.

2. Análise de Rede Social

A modelagem de sistemas em rede vem sendo aplicada em áreas diversas como epidemiologia (Moore e Newman, 2000) e colaboração cientifica (Newman, 2004). Uma rede pode ser representada por um grafo G=(V, E) formado por Vértices (V) e Arestas (E). Cada vértice ou nodo representa um ator e cada aresta representa a relação existente entre dois atores integrantes da rede. Uma rede pode ser direcionada ou não direcionada e as arestas podem ser valoradas ou não valoradas.

Segundo Katz e Martin (1997) a colaboração cientifica pode ser estudada segundo outros indicadores, porém a coautoria é o indicador mais utilizado. Logo, ao modelar uma rede de colaboração científica os vértices representam os autores e as arestas representam os artigos produzidos em parceria com outros autores. Esse tipo de rede é não direcionada e valorada porque a relação de coautoria é assíncrona e a valoração ocorre em função do número de artigos publicados em conjunto.

Dados modelados em rede são passíveis de serem analisados através de métricas de análise de rede social, a qual por sua vez, tem suas raízes na teoria de grafos. Segundo WASSERMAN e FAUST (1994) a área de análise de rede social (*social network analysis - sna*) tem atraído muito interesse nas últimas décadas. Através das métricas de *sna* é possível identificar aspectos, tais como: a) padrões de relacionamento entre os atores de uma rede; b) a conectividade entre os mesmos; c) a formação de *clusters*; d) a evolução da rede ao longo do tempo e, e) o fluxo de comunicação, informação e conhecimento dentro da rede.

Uma rede pode ser analisada segundo o escopo de estrutura, onde a medida de densidade é utilizada. Segundo Scott (2000), a densidade é um dos conceitos mais utilizados em teoria dos grafos, pois esta medida descreve o nível geral de ligações entre os pontos de um grafo. Um grafo "completo" é aquele em que todos os pontos são adjacentes um ao outro, ou seja, cada ponto é ligado diretamente a todos os outros pontos. Quanto mais pontos estão ligados uns aos outros, mais denso será o gráfico. No contexto de uma rede de coautoria a densidade reflete o percentual do total da rede com o qual um ator foi coautor de um artigo. (FISCHBACH; PUTZKE e SCHODER, 2011).

No escopo individual, existem algumas métricas de centralidade que procuram descrever as propriedades de localização de um ator na rede. Os atores mais importantes ou mais proeminentes estão normalmente localizados em posições estratégicas dentro da

rede (WASSERMAN; FAUST, 1994). A centralidade de um ator pode ser local ou global. A centralidade local está preocupada com a importância de um ator na sua vizinhança, enquanto que a centralidade global diz respeito a proeminência do ator dentro de toda a rede. A centralidade local é medida através da centralidade de grau (*degree centrality*), enquanto que a centralidade global é medida através da centralidade de intermediação (*betweeness centrality*) e centralidade de proximidade (*closeness centrality*).

A centralidade de grau de um ator corresponde ao número de arestas incidentes ou ao número de vértices adjacentes a ele. Segundo Freeman (1979) a centralidade de grau reflete a posição e o papel do ator em termos de popularidade e atividade. Em redes valoradas, onde a aresta possui um peso, a centralidade de grau pode levar em conta o valor ou peso da aresta. Em redes de coautoria essa medida determina o grau de colaboração de um ator.

A centralidade de proximidade é uma medida que indica a proximidade de um determinado ator em relação aos demais atores da rede, sendo definida pela soma das distâncias geodésicas entre um determinado vértice e todos os outros vértices do grafo (FREEMAN, 1979). Numa rede de coautoria, um autor com uma centralidade de proximidade alta pode indicar uma maior possibilidade de estabelecer parcerias de publicação na rede por estar mais próximo em relação a todos os outros autores (SOUZA; BARBASTEFANO; LIMA, 2012).

Por sua vez, a centralidade de intermediação mede o quanto um determinado ator se encontra "entre" os vários outros atores no grafo, ou seja, atribui importância a um ator em função do fluxo que passa por ele para interligar outros dois atores da rede através do menor caminho possível. Numa rede de coautoria, um autor com alto valor de centralidade de intermediação indica que um número significativo das parcerias estabelecidas na rede envolve, de forma direta ou indireta, as publicações relacionadas a esse ator (SOUZA; BARBASTEFANO; LIMA, 2012).

Uma das maiores preocupações de analistas de redes sociais é a identificação de subgrupos de atores dentro de uma rede. Subgrupos são subconjuntos de atores entre os quais existem laços fortes, diretos, intensos, frequentes ou positivos (WASSERMAN; FAUST, 1994). A identificação de componentes ou subgrafos dentro de uma rede é uma das técnicas para analisar uma rede do ponto de vista de um grupo de atores. Componentes são subgrafos que estão conectados dentro do grafo, mas desconectados entre os subgrafos. Se um grafo contém um ou mais pontos "isolados", esses pontos também são chamados de componentes. Componente gigante é o nome dado ao subgrafo que contem o maior numero de atores conectados. Numa rede de coautoria a presença de mais de um componente na rede indica a existência de grupos que publicam isoladamente

3. Procedimentos metodológicos

Os procedimentos dessa pesquisa estão divididos em três etapas:

1) Coleta de dados: A base de dados utilizada na pesquisa foi a Scopus. Ela é considerada a maior base de dados de resumos, citações e textos completos da literatura científica mundial revisada, com cobertura desde 1960, com mais de 20.500 títulos de

aproximadamente 5.000 editoras internacionais e atualizações diárias (SCOPUS, 2013). Para a pesquisa foram recuperados todos os documentos com o termo "ontolog*" produzidos por ao menos um autor afiliado a uma instituição brasileira. Para isso foi utilizado o termo "ontolog*" nos campos *title*, *abstract*, *keywords* juntamente com o termo "brazil" no campo *affiliation country*. A pesquisa retornou 1179 documentos, os quais foram exportados para o formato .RIS.

2) Normalização dos dados: A normalização dos dados foi realizada através de um processo semiautomático, com a extração e ordenação dos nomes de todos os autores através de uma aplicação e a conferência manual das inconsistências nos nomes dos autores.

3) Análise de rede: a rede de coautoria foi criada por uma aplicação que analisou e contabilizou todas as coautorias dos documentos a partir dos documentos coletados e normalizados. A rede consiste de uma relação de nodos e uma relação de pares de nodos juntamente com o peso (número de artigos publicados em coautoria). A análise da rede foi efetuada com a utilização do software de análise exploratória de dados Gephi (BASTIAN; HEYMANN; JACOMY, 2009).

4. Resultados

Na rede de coautoria foram identificados 2738 autores e 12345 relações. A densidade encontrada foi 0,003 onde o valor máximo é 1.0 e o grau médio de colaboração foi 9.0. Os autores foram classificados segundo a medida de centralidade de grau que leva em consideração o peso das relações (C.G. c/ Peso) e que determina o grau de colaboração entre os atores da rede. A tabela 1 apresenta dez autores, sua posição no ranking e o grau de colaboração.

Tabela 1: Ranking dos pesquisadores brasileiros em ontologia

Ranking	Instituição	Autor	Nu. Doc.	C.G.	C.G. c/Peso
2	UFRJ	De Souza# J.M.	20	38	85
3	UFES	Guizzardi# G.	42	40	75
14	PUCRIO	Casanova# M.A.	14	30	56
61	UFPE	Freitas# F.	16	43	55
62	UFC	De Macedo# J.A.F.	8	28	54
119	UFAL	Bittencourt# I.I.	9	29	46
121	UFRJ	Xexeo# G.B.	11	23	45
122	UFAL	Costa# E.	9	26	44
135	UFC	Vidal# V.M.P.	7	17	40
139	UNICAMP	Medeiros# C.B.	17	22	38

A análise de rede revelou a existência de 348 componentes isolados, onde o maior componente (componente gigante) possui 941 autores e representa 34,37% da rede. A figura 01 apresenta o componente gigante com destaque para os autores com maior centralidade de intermediação, tais como Siqueira# S.W.M. (0,35), De Souza#

J.M. (0,32) e Breitman# K.K. (0,31). Em relação à centralidade de proximidade os autores com maiores graus são Siqueira# S.W.M., De Souza# J.M. e Carvalho, G., todos com 0.14.

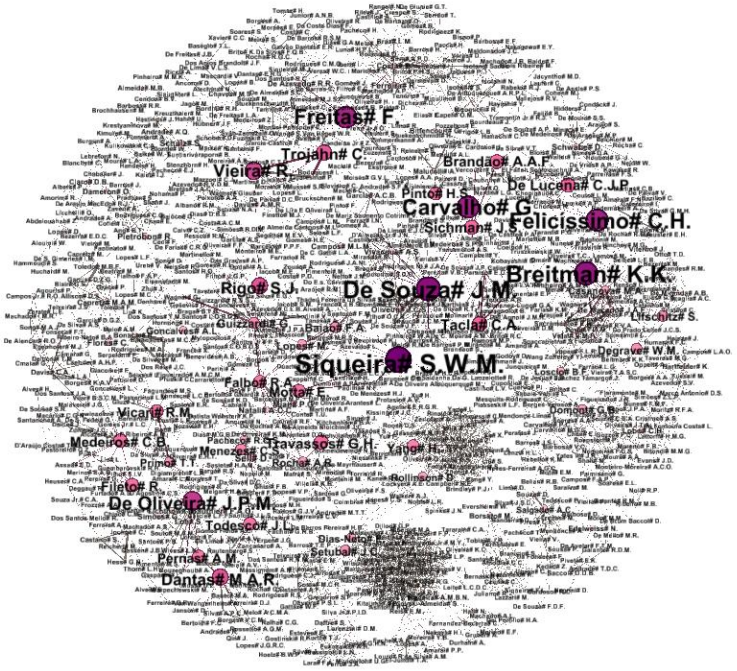


Figura 1: Componente gigante da rede de coautoria de pesquisadores em ontologia

O segundo maior componente encontrado na rede possui 110 autores e corresponde a 4% da rede. Esse componente foi formado principalmente pela publicação de cinco artigos que abordam o uso de ontologias na área de Bioinformática e foram coautorados por pesquisadores de diversos países. O terceiro e quarto maior componente, respectivamente com 59 e 54 autores, também são formados por autores com publicações nessa área. O quinto e o sexto componente contam com 51 e 44 autores respectivamente, cujas publicações são na área de ciência da computação.

5. Discussão e Conclusão

O ranking original dos autores com maior grau de colaboração apresentou muitos autores da área de Bioinformática. Verificou-se que os trabalhos desses autores abordam apenas o uso de ontologias conhecidas na área. Como os trabalhos não contribuem diretamente para o avanço da pesquisa em ontologia e o alto grau de colaboração encontrado nesses autores está mais relacionado com a coautoria de poucos documentos com muitos outros autores optou-se por não apresentá-los na tabela 1.

No ranking original de colaboração somente dois autores da área de Computação ocupam as primeiras dez posições. Isso pode evidenciar a necessidade de maior colaboração entre os pesquisadores dessa área.

A quantidade de componentes isolados (348) quando comparada ao numero de atores da rede (2738) indica que a rede de coautoria dos pesquisadores brasileiros em

ontologia é bem fragmentada, ou seja, existem de grupos de pesquisadores trabalhando isoladamente ou sem colaboração entre grupos. A análise do segundo ao décimo componente isolado indica uma polarização das pesquisas entre as áreas de Computação e Bioinformática.

O maior componente de rede encontrado indica que 1/3 dos atores então conectados por algum caminho. Nesse componente gigante, pesquisadores da área de Computação lideram o ranking de colaboração.

Esse trabalho apresentou uma análise preliminar da rede de pesquisadores brasileiros em ontologia, a partir da qual se pode concluir que existe espaço para uma maior colaboração entre os grupos de pesquisadores brasileiros e que as áreas de conhecimento mais representativas envolvidas com a pesquisa em ontologia são Computação e Bioinformática.

6. Referências

BASTIAN, M.; HEYMANN, S.; JACOMY, M. Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media, 2009.

FREEMAN, L. Centrality in Social Networks: Conceptual Clarification. Social Networks, 1(3), 215–239, 1979.

FISCHBACH, K.; PUTZKE, J.; SCHODER, D. Co-authorship networks in electronic markets research. Electronic Markets, 21(1), 19–40, 2011.

KATZ, J. S.; MARTIN, B. R. What is research collaboration? Res. Policy, 26(1):1-18, 1997.

MOORE, C.; NEWMAN, M. E. J. Epidemics and percolation in small-world networks. Phys. Rev. E 61, 5678–5682, 2000.

NEWMAN, M. E. Coauthorship networks and patterns of scientific collaboration. Proceedings of the National Academy of Sciences of the United States of America, v. 101, n. Suppl 1:5200-5205, april 2004.

SCOPUS. Base de dados de literatura de pesquisa. Disponível em: http: http://www.scopus.com

SCOTT, J. Social Network Analysis. A Handbook. 2nd edition. SAGE Publications: London, 2000.

SOUZA, C. G.; BARBASTEFANO, R. G.; LIMA, L. S. Redes de colaboração científica na área de química no Brasil: um estudo baseado nas coautorias dos artigos da revista Química Nova. Química Nova, São Paulo, v. 35, n. 4, p. 671-676, 2012.

WASSERMAN, S.; FAUST, K. Social Network Analysis: methods and applications. Cambridge University Press. Structural analysis in social the social sciences series, v. 8, (1994) 1999. 857 p. ISBN 0-521-38707-8.

Ontologias Aplicadas ao Problema de Correlação Litológica no Domínio da Geologia do Petróleo*

Luan Fonseca Garcia, Joel Luis Carbonera, Mara Abel

[1] Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

{lfgarcia,jlcarbonera,marabel}@inf.ufrgs.br

Abstract. *In this work we apply a domain ontology for developing a computational approach for the task of lithologic correlation, within the Petroleum Geology domain. In this context, a domain ontology is applied for imposing a rich and homogeneous structure to the visual descriptions of the domain objects that are the targets of this task. In our approach, we combine the use of ontologies with clustering techniques and sequence alignment algorithms, which are typically applied in DNA sequencing. A domain ontology with a vocabulary sufficiently expressive for allowing rich visual descriptions of the domain objects is a key aspect of our proposal.*

Resumo. *Neste trabalho, exploramos o uso de uma ontologia de domínio para o desenvolvimento de uma abordagem computacional para a tarefa de correlação litológica, no domínio da Geologia do Petróleo. Neste contexto, uma ontologia de domínio é utilizada para impor uma estrutura rica e homogênea às descrições visuais dos objetos de domínio que são o foco desta tarefa. Além da ontologia de domínio, a abordagem também combina técnicas de clusterização e algoritmos de alinhamento de sequências, tipicamente utilizados para realizar o sequenciamento de DNA. A disponibilidade de uma ontologia, com um vocabulário expressivo o suficiente para proporcionar uma descrição visual rica dos objetos do domínio é um aspecto chave desta proposta.*

1. Introdução

Domínios visuais são aqueles em que a resolução de problemas é fortemente baseada na aplicação de *conhecimento visual* dos especialistas. Consideramos conhecimento visual como sendo o conjunto de modelos mentais que suportam o processo de raciocínio sobre informação relacionada ao arranjo espacial e outros aspectos visuais das entidades de domínio [Lorenzatti et al. 2009, Carbonera et al. 2011]. Este trabalho insere-se no contexto do projeto *Obaitá*, desenvolvido pelo grupo *BDI* (grupo de bancos de dados inteligentes da UFRGS). Neste projeto, investigamos abordagens integradas para aquisição, modelagem, representação e raciocínio sobre conhecimento visual. Um dos resultados esperados é uma ontologia para o domínio (visual) da *Estratigrafia Sedimentar*, que viabilize o desenvolvimento de diversos sistemas baseados em conhecimento, que operem sobre uma mesma conceitualização deste domínio. Em [Lorenzatti et al. 2009] são apresentados os passos iniciais em direção a este resultado, enquanto em [Carbonera 2012] esta ontologia é expandida, utilizando-se a abordagem descrita em [Carbonera et al. 2012].

*Este trabalho foi desenvolvido com recursos do CNPq e do programa PRH PB-217, mantido pela Petrobras e Agência Nacional do Petróleo (ANP). Também gostaríamos de agradecer à Endeeper pela disponibilização dos dados utilizados para a realização do trabalho.

Atualmente, investigamos abordagens para a tarefa de correlação litológica, no domínio da Estratigrafia Sedimentar, que se beneficiem da ontologia desenvolvida. A Estratigrafia Sedimentar é uma sub-área da Geologia que estuda as camadas que compõem a Terra e busca determinar como ocorreu a formação dessas camadas. Neste domínio, na tarefa de correlacão litológica o geólogo busca reconhecer a mesma fácies sedimentar (Figura 1) em duas ou mais seções estratigráficas diferentes, mesmo que espacialmente distantes entre si. Ou seja, nesta tarefa o geólogo investiga a continuidade lateral de fácies sedimentares em subsuperfície, onde não é possível realizar observação direta destas unidades. A correlação permite determinar a distribuição espacial e o volume das rochas que subsidiam a avaliação de economicidade dos reservatórios de petróleo.

Para alcançar este objetivo, o geólogo inicia descrevendo visualmente *corpos de rocha*, tal como o *testemunho de sondagem* apresentado na Figura 1. A descrição destes corpos envolve discretizá-los em *fácies sedimentares* e descrever todos os atributos visuais que caracterizam cada uma delas. Neste contexto, a fácies sedimentar é uma dada porção de um corpo de rocha, visualmente distinguível das porções adjacentes. Além dos atributos que as caracterizam, as fácies sedimentares possuem uma ou mais *estruturas sedimentares*, que correspondem a padrões geométricos externamente visíveis, que indicam padrões de arranjos espaciais internos dos grãos que constituem uma fácies.

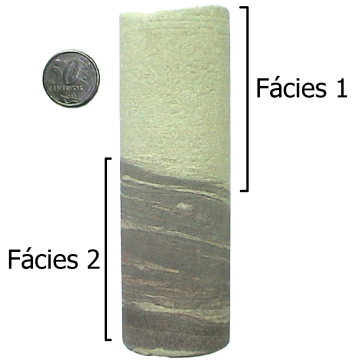


Figura 1. Trecho de testemunho de sondagem, com duas fácies distintas. Adaptado de [Lorenzatti 2009].

A continuidade das unidades de rochas é julgada pela similaridade entre fácies, visto que elas são as porções discretizadas de rocha passíveis de observação direta e visualmente distinguíveis das demais. A correspondência entre corpos de rocha pode ser parcial, assim como a similaridade entre as unidades relacionadas; como exemplificado na Figura 2, onde a fácies 4 está presente nas seções A e B, mas não está presente na seção C. É importante notar que, a correlação é estabelecida não apenas com base na identificação do mesmo tipo de rocha nos diferentes poços, mas principalmente pelo sequenciamento semelhante de diferentes tipos de rochas em cada um dos poços.

Atualmente, a correlação é realizada a partir de múltiplos registros textuais distintos, sem uma estrutura padrão, capturados por geólogos diferentes, sem apoio de um vocabulário padrão. Essas condições fazem com que, no atual estado da arte, a tarefa de correlação litológica careça de métodos automáticos para processamento em larga escala. Neste trabalho, nós propomos que a correlação pode ser realizada com métodos computacionais de correlação automáticos, desde que as descrições de rochas sejam orientadas

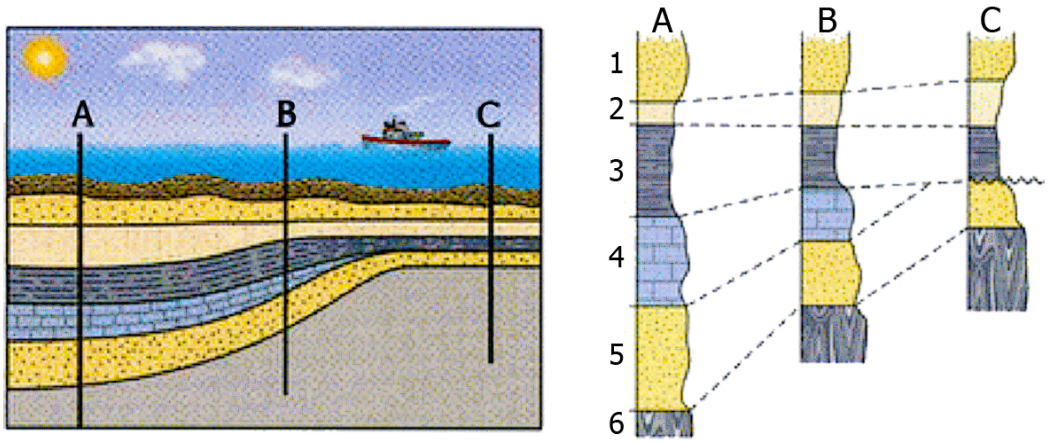


Figura 2. Representação de uma correlação litológica entre três seções estratigráficas distintas (A, B e C), envolvendo seis fácies. Adaptada de [Parsons 2013].

por uma ontologia de domínio bem fundamentada, tal como a incorporada no software Strataledge®[1] e que é apresentada em detalhes em [Carbonera 2012]. A utilização de uma ontologia de domínio expressiva, permite impor uma estrutura formal homogênea às descrições dos objetos de domínio, viabilizando a descrição de um conjunto rico de informações, através de um vocabulário formal bem definido. Essas informações, capturadas de forma uniforme e não ambígua, permitem a comparação entre porções descritas das unidades de rocha espacialmente distintas, que não são influenciadas pelo uso de diferentes vocabulários e estilos descritivos, suportando a correlação entre elas. Além disso, considerando que a ontologia especifica a conceitualização compartilhada no domínio, o seu uso para descrição dos objetos do domínio permite que os sistemas processem as informações acerca desses objetos de um modo que se aproxime da forma como os geólogos os concebem. Assim, partimos da hipótese de que abordagens automáticas para correlação podem se beneficiar do uso de ontologias, oferecendo resultados geologicamente mais significativos.

2. Abordagem proposta

Diversas abordagens têm sido propostas para lidar computacionalmente com o problema da correlação litológica. Uma abordagem que tem se revelado promissora neste sentido, tal como a adotada em [Waterman and Raymond Jr 1987], envolve a aplicação de *algoritmos de alinhamento de sequências*. Estes algoritmos vêm sendo utilizados com sucesso na tarefa de alinhamento de sequências de DNA no domínio da bioinformática. Entre estes algoritmos, destaca-se o algoritmo de programação dinâmica *Smith-Waterman*, que possui resultado ótimo para o alinhamento de seqüências locais.

Segundo [Chao and Zhang 2009], o algoritmo de Smith-Waterman parte de uma sequência $A = a_1 a_2 ... a_m$ e uma sequência $B = b_1 b_2 ... b_n$, que podem ter tamanhos diferentes. De modo geral, o alinhamento entre estas sequências é obtido pela inserção de lacunas (representadas pelo caractere "-") em ambas, representando deslocamentos entre os segmentos similares, de tal modo que o tamanho final de ambas seja idêntico,

[1] `http://www.endeeper.com/products/software/strataledge`

sendo que não pode haver alinhamentos entre lacunas. A Figura 3 apresenta dois exemplos de alinhamentos de sequências de DNA resultantes da aplicação deste algoritmo. O funcionamento detalhado deste algoritmo foge ao escopo deste artigo, mas pode ser encontrado em [Chao and Zhang 2009].

```
-ATACATGTC--T        -----AATGCCATTGAC----GG
G-TAC--GTCGG-        CAGCC--T--C---G-CTTAG--
     (a)                    (b)
```

Figura 3. Dois exemplos (a e b) de alinhamentos de pares de sequências de DNA realizados pelo algoritmo de Smith-Waterman.

Para aplicar o algoritmo de Smith-Waterman sobre duas sequências, deve haver uma maneira de comparar elementos de ambas, determinando quando eles são equivalentes. Quando aplicado no alinhamento de sequências de DNA, este algoritmo opera sobre strings construídas a partir de um alfabeto finito – as quatro letras que representam as bases nitrogenadas básicas do DNA. Neste caso a comparação entre elementos de duas sequências é trivial, bastando verificar se os dois elementos são a mesma letra (representando o mesmo tipo de base nitrogenada). Por outro lado, quando aplicado ao problema de correlação litológica, o algoritmo deve ser capaz de operar sobre sequências de fácies sedimentares. Em relação a este ponto, [Griffiths and Bakke 1990] afirma que aplicações convencionais deste algoritmo para o problema em foco devem determinar uma forma de codificar a informação das fácies sedimentares de um modo análogo ao que ocorre no caso do sequenciamento de DNA, utilizando um conjunto finito de símbolos bem definidos, que podem ser comparados pelo algoritmo. Em nossa abordagem, adaptamos o algoritmo para que sejam comparados os clusters aos quais as duas fácies comparadas pertencem. Para isso, antes de aplicar o algoritmo de alinhamento, utilizamos um algoritmo de clusterização[2] sobre um *dataset* formado pelo conjunto de descrições das fácies que se pretende correlacionar. Lembrando que estas descrições são realizadas com suporte da ontologia de domínio. A partir desde passo, é obtido um modelo *clusterizador* que é capaz de classificar instâncias de fácies (incluindo instâncias não consideradas durante o treinamento do clusterizador). Este clusterizador, por sua vez, é utilizado pelo algoritmo de alinhamento para comparar se duas fácies pertencem à um mesmo *cluster*. Se as duas fácies, em corpos de rocha distintos, estão no mesmo cluster, consideramos que elas são equivalentes e que podem ser alinhadas. Atualmente, o treinamento do clusterizador é realizado através da API Weka[3], aplicando o algoritmo *EM* (*expectation–maximization*) [Witten et al. 2011]. A Figura 4 representa esquematicamente a abordagem proposta.

A ontologia de domínio utilizada [Carbonera 2012] descreve o conceito de fácies através de 23 atributos. Na fase de *conversão* das descrições de corpos de rocha para o dataset de treinamento do clusterizador (Figura 5), cada instância f do conceito fácies na ontologia é convertida em um vetor de características v. Cada posição p neste vetor representa um atributo descritivo a que caracteriza o conceito de fácies na ontologia, de modo que cada valor v_p do vetor v representa o valor específico que f possui para o respectivo

[2]De acordo com [Witten et al. 2011], clusterização é uma técnica de mineração de dados utilizada para determinar um conjunto de categorias ou agrupamentos (*clusters*) a partir de um conjunto de dados sem classificação prévia.

[3]`http://www.cs.waikato.ac.nz/ml/weka/`

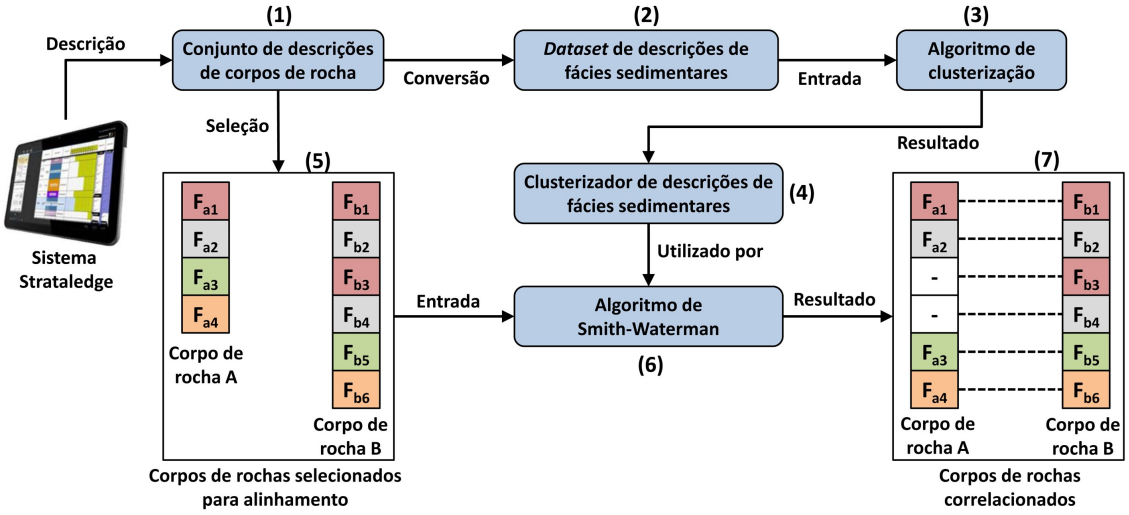


Figura 4. Representação dos procedimentos realizados na abordagem proposta. Cada F_{ij} representa uma fácies j no corpo de rocha i.

atributo. Além disso, em nosso caso, o vetor v também possui uma posição especial que representa a relação $temEstrutura$ entre a instância de fácies e uma instância de estrutura sedimentar. Esta posição especial recebe como valor o tipo específico da instância e de estrutura sedimentar relacionada à fácies f. Assim, o dataset de treinamento é um conjunto V de vetores de características v, cada qual representando uma instância f do conceito de fácies, descrita pela ontologia.

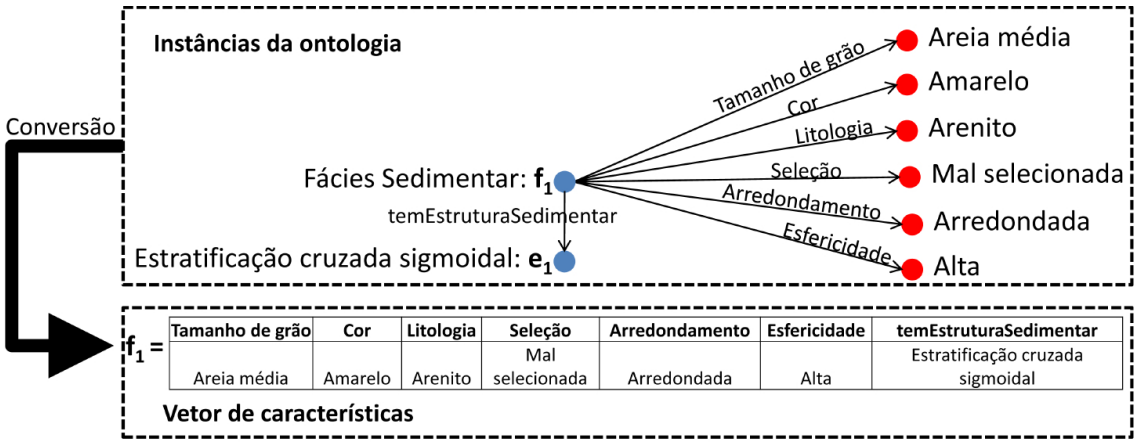


Figura 5. Representação do processo de conversão de instâncias de fácies sedimentares em vetores de características, considerando um conjunto reduzido de atributos.

3. Considerações finais

Neste trabalho, apresentamos uma abordagem computacional para correlação litológica automática no domínio da Estratigrafia Sedimentar. Esta abordagem está alinhada às abordagens mais promissoras oferecidas pela literatura. A nossa principal contribuição reside no uso de uma ontologia de domínio para conferir uma estrutura formal homogênea às descrições dos objetos do domínio. Assim, considerando que a ontologia captura de

modo formal e explícito a conceitualização compartilhada pela comunidade, ela permite que os usuários descrevam os objetos do domínio de um modo padrão, formal e com uma estrutura rica de informações. Isto viabiliza o tratamento computacional destas descrições e permite que nossa abordagem processe informações sobre os objetos do domínio de um modo que se aproxime da forma como os geólogos os conceitualizam.

Na fase atual deste trabalho, com o auxílio de especialistas do domínio, estamos coletando um conjunto de descrições de corpos de rocha reais. Nos próximos passos deste projeto será investigado como considerar a importância relativa dos atributos da fácies durante a clusterização, de um modo que seja possível determinar similaridades geologicamente mais significativas entre fácies sedimentares.

Referências

Carbonera, J., Abel, M., dos Santos Scherer, C. M., and Bernardes, A. (2012). Abordagem para aquisição de conhecimento visual e refinamento de ontologias para domínios visuais. In Vieira, R., Guizzardi, G., and Fiorini, S. R., editors, *Proceedings of Joint V Seminar on Ontology Research in Brazil and VII International Workshop on Metamodels, Ontologies and Semantic Technologies*, volume 776.

Carbonera, J. L. (2012). Raciocínio sobre conhecimento visual: Um estudo em estratigrafia sedimentar. Master's thesis, Universidade Federal do Rio Grande do Sul (UFRGS).

Carbonera, J. L., Abel, M., Scherer, C. M. S., and Bernardes, A. K. (2011). Reasoning over visual knowledge. In Vieira, R., Guizzardi, G., and Fiorini, S. R., editors, *Proceedings of Joint IV Seminar on Ontology Research in Brazil and VI International Workshop on Metamodels, Ontologies and Semantic Technologies*, volume 776.

Chao, K.-M. and Zhang, L. (2009). *Sequence comparison: theory and methods*, volume 7. Springer.

Griffiths, C. and Bakke, S. (1990). Interwell matching using a combination of petrophysically derived numerical lithologies and gene-typing techniques. *Geological Society, London, Special Publications*, 48(1):133–151.

Lorenzatti, A. (2009). Ontologia para domínios imagísticos: Combinando primitivas textuais e pictóricas. Master's thesis, UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL.

Lorenzatti, A., Abel, M., Nunes, B. R., and Scherer, C. M. S. (2009). Ontology for imagistic domains: Combining textual and pictorial primitives. In Heuser, C. A. and Pernul, G., editors, *ER Workshops*, volume 5833 of *Lecture Notes in Computer Science*, pages 169–178. Springer.

Parsons, S. B. (2013). Historical geology. Disponível em: <http://www.ocean.odu.edu/~spars001/geology_112/laboratory/session_04/handout.html>. Acesso em: 14 de Julho de 2013.

Waterman, M. S. and Raymond Jr, R. (1987). The match game: new stratigraphic correlation algorithms. *Mathematical geology*, 19(2):109–127.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier.

Uma Ontologia para Padronização do Domínio de Robótica e Automação

Sandro Rama Fiorini, Joel Luis Carbonera, Vitor A. M. Jorge, Edson Prestes, Mara Abel

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{srfiorini,jlcarbonera,vamjorge,prestes,marabel}@inf.ufrgs.br

Abstract. *This paper summarizes the development of a core ontology in the Robotics and Automation domain (R&A), as part of the efforts of IEEE RAS to standardize the field. Tasks and interaction in which robots find themselves in are increasing in complexity. That imposes the requirement for a formally specified body of knowledge that is necessary in such processes. In this context, we propose a core ontology that describes the basic concepts and relations encompassing the R&A domain, based on other existent standard vocabularies and expert knowledge.*

Resumo. *Este artigo sumariza o desenvolvimento de uma ontologia de núcleo para o domínio de Robótica e Automação (R&A) como parte do esforço da IEEE RAS para padronização da área. O aumento da complexidade das tarefas e interações realizadas por robôs coloca a necessidade de um padrão que especifique formalmente o conhecimento necessário nestes processos. A ontologia de núcleo proposta captura os principais conceitos e relações abrangendo o domínio de R&A como um todo, usando como base padronizações já existentes na literatura da área e conhecimento especialista.*

1. Introdução

O aumento constante da complexidade das tarefas realizadas por robôs tem demandando mecanismos mais sofisticados de colaboração entre eles e outros agentes, sejam estes outros robôs ou humanos. Neste contexto, torna-se evidente a necessidade de um *padrão* que capture de forma explícita e formal o conhecimento compartilhado no campo da robótica e automação (R&A). A existência de um padrão que defina precisamente os conceitos neste domínio é fundamental para promover a interoperabilidade semântica entre os diversos agentes e sistemas envolvidos. Neste cenário, ontologias têm sido adotadas como uma tecnologia capaz de promover esta interoperabilidade, uma vez que elas constituem especificações formais de conceitualizações compartilhadas (Studer, Benjamins, & Fensel, 1998). A utilização de ontologias em processos de padronização tem sido explorada na linha de pesquisa em padrões baseados em ontologias (*ontology-based standards*). Iniciativas recentes neste sentido, como a *Ontology-based Standards Initiative*[1], tem promovido uma aproximação e a troca de experiências entre a comunidade de pesquisa em ontologias e a comunidade de padronização, enfatizando como o

[1] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologyBasedStandards

processo de padronização pode ser auxiliado por princípios, ferramentas e metodologias tipicamente relacionados ao desenvolvimento de ontologias.

Desde Novembro de 2011, nosso grupo, chamado Ontologies for Robotics and Automation Working Group (ORA WG), vem atuando como um grupo de trabalho oficial junto à IEEE-SA Standards Board, assumindo o objetivo de padronizar a representação de conhecimento no domínio da robótica. Este grupo, que inclui mais de 140 pessoas de mais de 20 países, tem trabalhado ativamente com instituições da indústria, academia e governo para desenvolver um conjunto de ontologias, ferramentas e metodologias associadas, para serem usadas como um padrão no domínio da Robótica e Automação (R&A).

O ORA WG é composto por quatro subgrupos: *Industrial Robots* (InR), *Service Robots* (SeR), *Autonomous Robots* (AuR) and *Upper Ontology/Methodology* (UpOM). Os três primeiros são responsáveis pela elaboração de ontologias para três subdomínios da R&A considerados nesta fase do projeto, sendo eles robótica industrial, de serviço e autônoma, respectivamente. Já o UpOM tem como principal objetivo o desenvolvimento de uma *ontologia de núcleo* (*core ontology*) que especifique os conceitos mais gerais do domínio, desempenhando o papel de base para a integração consistente de todas as subontologias desenvolvidas no projeto. Além disso, o UpOM também está encarregado de avaliar e integrar as subontologias propostas pelos demais subgrupos do ORA WG.

Este artigo descreve a *Ontologia de Núcleo para Robótica e Automação* (*Core Ontology for Robotics and Automation*, ou CORA), desenvolvida pelo UpOM e detalhada em (Prestes et al., 2013); e apresenta os primeiros resultados da axiomatização do modelo. Primeiramente discutimos o processo de engenharia de ontologias realizado durante o desenvolvimento da ontologia CORA. Em seguida, apresentamos os seus principais conceitos, com ênfase para o conceito de robô.

2. Desenvolvimento da Ontologia

Devido à complexidade esperada em um projeto como este, é necessário determinar quais metodologias, ferramentas e princípios serão utilizados para guiar o processo de desenvolvimento das ontologias. Adotamos a METHONTOLOGY (Lopez, Perez, & Juristo, 1997) como metodologia geral de desenvolvimento de ontologias, uma vez que ela oferece características ajustadas às necessidades do projeto, tais como a independência de aplicação e a ênfase no desenvolvimento de ontologias no nível do conhecimento. A METHONTOLOGY também estabelece um conjunto de atividades que devem ser realizadas durante o desenvolvimento, especifica o ciclo de vida da ontologia ao longo do desenvolvimento e indica técnicas para realizar cada atividade proposta no ciclo de vida. Além disso, adotamos a OntoClean (Guarino & Welty, 2009) como ferramenta de avaliação da ontologia em desenvolvimento. Também utilizamos as meta-propriedades oferecidas pela OntoClean como princípios para avaliação de outras fontes de conhecimento consultados para a elaboração da ontologia proposta. Finalmente, adotamos uma abordagem *middle-out* para identificação de conceitos, isto é, identificando antes aqueles mais relevantes e em seguida os mais abstratos e os mais específicos.

O processo de desenvolvimento foi iniciado com a identificação de fontes das quais o conhecimento de domínio seria adquirido. As principais fontes identificadas foram: padrões já existentes no domínio; livros-texto, artigos revisados por pares; espe-

cialistas no domínio; e ontologias já existentes, incluindo não apenas ontologias de domínio, mas também ontologias de topo, como a SUMO (Niles & Pease, 2001).

Analisando as fontes identificadas, constatamos que as ontologias já desenvolvidas no domínio, em geral, focam apenas em um subconjunto restrito de conceitos do domínio, adotando significados específicos, compartilhados apenas por pequenos grupos de pessoas. Devido a isto, decidimos iniciar a identificação de conceitos a partir do documento ISO 8373:2012, que define em linguagem natural termos genéricos que são comuns no domínio da R&A. Este documento foi considerando uma fonte inicial de conhecimento adequada, uma vez que este padrão foi elaborado com o intuito de estabelecer um acordo inicial na comunidade de R&A.

A partir deste documento da ISO, foram identificados os termos e definições relacionados aos conceitos mais gerais do domínio e que deveriam participar da ontologia de núcleo. Este conhecimento então foi estruturado e representado usando as representações intermediárias previstas pela METHONTOLOGY. Neste estágio também realizamos uma avaliação da conceitualização capturada do documento da ISO, utilizando a OntoClean. Este processo permitiu constatar que as definições oferecidas pela ISO para conceitos chave são ambíguas e que alguns dos compromissos ontológicos não estão explícitos nas definições, permitindo algumas interpretações não pretendidas. Detalhes desta análise são apresentados em (Prestes et al., 2013).

O próximo passo foi a coleta de definições alternativas para os termos identificados anteriormente. A partir de uma análise das definições alternativas, foram elaboradas novas definições para estes termos, com o intuito de contemplar os principais aspectos enfatizados pelas definições encontradas.

Finalmente, integramos a ontologia com uma ontologia de topo. Ontologias de topo ajudam a organizar a estrutura básica de ontologias mais específicas ao estruturar as principais categorias gerais, presentes em qualquer domínio. Nesta etapa, selecionamos a SUMO (Niles & Pease, 2001), uma ontologia de topo desenvolvida por um grupo de trabalho oficial da IEEE, que inclui colaboradores de diversas áreas. A SUMO oferece uma descrição flexível das categorias de topo e inclui as principais noções e distinções necessárias para a ontologia de núcleo de R&A.

3. Ontologia de núcleo para R&A

A CORA (Figura 1) é naturalmente uma ontologia sobre robôs e conceitos relacionados. O objetivo é descrever as qualidades que caracterizam robôs em geral. Ela descreve quatro categorias abrangentes: parte de robô, robô, grupo de robôs e sistema robótico. Dadas as restrições de espaço, apresentaremos uma breve descrição de cada conceito.

Talvez existam tantas definições para o termo "robô" quanto existem autores escrevendo sobre o assunto. Essa ambiguidade inerente ao termo se torna um problema quando se pretende defini-lo de forma abrangente. Por isso, decidimos definir "robô" somente em termos de condições *necessárias*. Isso nos permite cobrir todas as entidades que a comunidade considera como sendo um robô, ao custo de permitir a caracterização de entidades que não são usualmente consideradas como robôs por alguns roboticistas. Não obstante, os conceitos da nossa ontologia podem ser especializados para contemplar entidades com significado mais restrito, de acordo com as necessidades de subdomínios e aplicações de R&A.

Mais importante, decidimos por uma definição de robô que enfatiza os seus aspectos funcionais. Definimos robôs como dispositivos agentivos em um sentido amplo, capazes de agir no mundo físico com o propósito de completar uma ou mais tarefas. Em alguns casos, as ações de um robô podem estar subordinadas às ações de outros agentes, tais como agentes de software ou humanos. Um robô é composto por partes mecânicas e eletrônicas apropriadas. Robôs podem formar grupos sociais, onde eles interagem para atingir um objetivo único. Um robô (ou um grupo de robôs) pode formar sistemas robóticos junto com equipamentos situados no ambiente que facilitam o seu trabalho.

Um robô é um *dispositivo* e um *agente* no sentido da SUMO:

$$\forall x\ \text{Robô}(x) \rightarrow \text{SUMO:Agente}(x) \wedge \text{SUMO:Dispositivo}(x).$$

De acordo com ela, um dispositivo é um artefato (i.e. um objeto físico produto de fabricação), cujo propósito é participar como um instrumento em um processo. A SUMO define agente como "algo ou alguém que pode agir por si próprio e produzir mudanças no mundo." Robôs realizam tarefas agindo no ambiente ou em si mesmos. Ação é fortemente relacionada à agência, no sentido de que a ação define o agente.

Naturalmente, dispositivos podem ter partes. Definimos um conceito específico que captura a noção de parte de robô:

$$\begin{aligned}\forall x\ \text{ParteDeRobô}(x) \rightarrow\ & \text{SUMO:Dispositivo}(x) \\ & \wedge \exists y\ [\text{Robô}(y) \wedge \text{SUMO:componente}(x, y)],\end{aligned}$$

onde a relação $\text{SUMO:componente}(x, y)$ é uma relação partonômica que especifica que o objeto *x* é parte do objeto *y*; i.e. $\forall x\ \text{SUMO:componente}(x) \rightarrow \text{SUMO:parte}(x, y)$. É importante frisar que não assumimos a existência de algum dispositivo que é necessariamente parte de robô. A razão disso é que até mesmo os dispositivos mais especializados para robótica podem ser utilizados em dispositivos diferentes de robô. Isto caracteriza ParteDeRobô como um *papel formal*, no sentido de Guarino e Welty (2000); o conceito caracteriza qualquer outro dispositivo que pode compor um robô, de porcas e parafusos até manipuladores e atuadores.

Robôs podem também formar *grupos de robôs*. De acordo com a SUMO, um *grupo* é uma coleção de agentes. Um grupo de robôs é definido como:

$$\forall x\ \text{GrupoDeRobôs}(x) \rightarrow \text{SUMO:Grupo}(x) \wedge \forall y\ [\text{SUMO:membro}(y, x) \rightarrow \text{Robô}(y)].$$

A SUMO define grupo como sendo também um agente; sua agência emerge dos participantes do grupo. Esse conceito pode ser usado para descrever entidades tais como times de robôs, ou mesmo robôs complexos formados por diversos agentes robóticos independentes atuando em uníssono.

Robôs podem participar de *sistemas robóticos*. Um sistema robótico é composto por um robô ou um grupo de robôs, mais dispositivos situados no ambiente que dão suporte a atuação dos robôs. Dessa forma,

$$\begin{aligned}\forall x\ & \text{SistemaRobótico}(x) \\ & \rightarrow \exists y\ [\text{SUMO:Dispositivo}(y) \wedge \neg\text{Robô}(y) \wedge \text{SUMO:componente}(y, x) \\ & \qquad \wedge \text{suporta}(y, x)] \\ & \wedge \exists!\, z\ \big[\text{parte}(z, x) \wedge [\text{Robô}(z) \vee \text{GrupoDeRobôs}(z)]\big];\end{aligned}$$

Dado que Robô e Grupo de Robôs são conceitos disjuntos, o axioma anterior induz a presença de dois subconceitos de Sistema Robótico: (a) o Sistema Robótico Simples, que tem como parte um e somente um robô; e o Sistema Robótico Coletivo, que tem como parte um grupo de robôs. Dessa forma, excluímos a existência de sistemas robóticos com diversos robôs não integrados. Já um ambiente robótico é um ambiente equipado com um sistema robótico:

$$\forall x\ \text{AmbienteRobótico}(x) \rightarrow \text{Ambiente}(x) \\ \wedge\ \exists y\ [\text{SistemaRobótico}(y) \wedge \text{equipadoCom}(x, y)];$$

tal que os equipamentos que dão suporte ao sistema robótico estão obrigatoriamente localizados no ambiente; i.e.,

$$\forall x\ \forall y\ \text{equipadoCom}(x, y) \rightarrow \forall z\ [\text{suporta}(z, y) \rightarrow \text{SUMO:localizado}(z, x)]$$

Um bom exemplo de sistema robótico é uma célula de montagem em uma fábrica de automóveis. Ela é formada por um grupo de robôs industriais que montam os automóveis, mais todo o equipamento de apoio à montagem, como sensores, guindastes e esteiras de transporte.

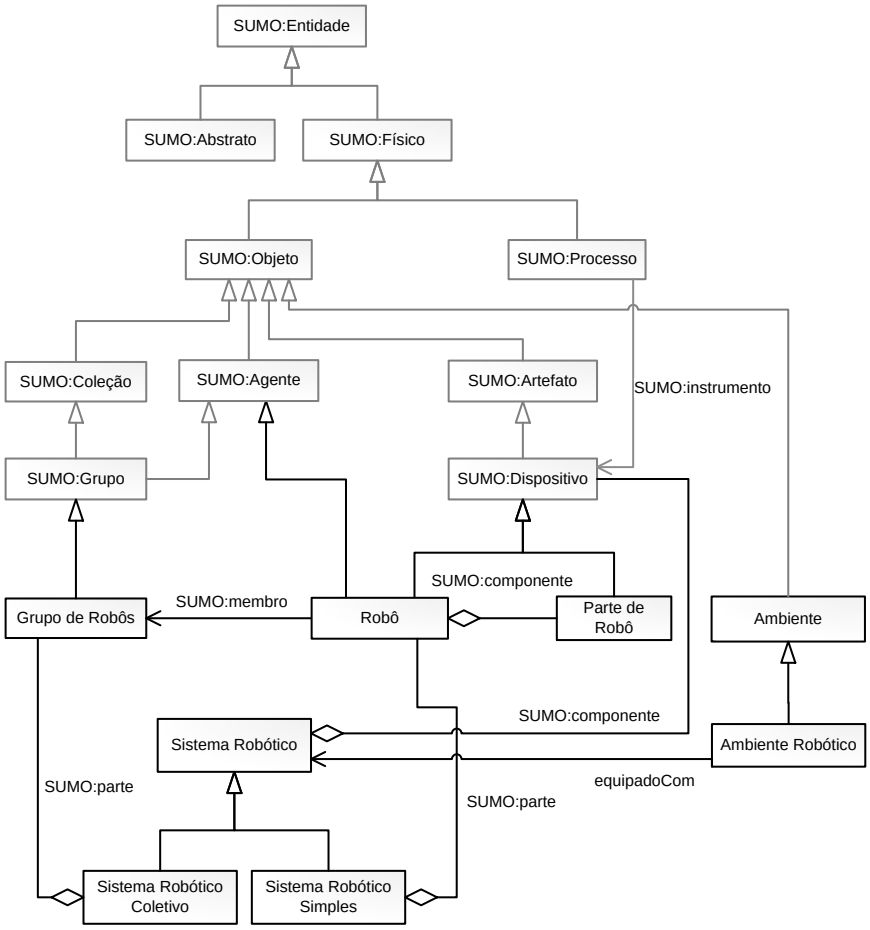


Figura 1. Visão geral da ontologia pra robótica e automação. Conceitos e relações em cinza são estabelecidos pela SUMO.

4. Considerações Finais

Ainda existe trabalho a ser feito para que o ORA WG chegue a um conjunto de ontologias padrão para R&A. Atualmente, temos duas frentes de trabalho. Estamos estendendo a CORA para especificar outros aspectos importantes do domínio, como noções de posicionamento, tarefas e estado do robô (e.g. Carbonera et al., 2013). Além disso, estamos trabalhando com os demais subgrupos para o desenvolvimento da ontologia resultante do projeto, que integra as diversas ontologias específicas com a ontologia de núcleo. No futuro próximo pretendemos fundamentar as ontologias propostas pelo UpOM de acordo com a ontologia de fundamentação UFO (Guizzardi, 2005).

5. Agradecimentos

Os autores gostariam de agradecer a CNPq, CAPES, ANP e projeto PRH PB-217 pelo suporte financeiro a este trabalho.

6. Referências

Carbonera, J. L., Fiorini, S. R., Prestes, E., Jorge, V. A. M., Abel, M., Madhavan, R., … Schlenoff, C. (2013). Defining Position in a Core Ontology for Robotics. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*. Tokyo, Japan. (Aceito para publicação).

Guarino, N., & Welty, C. A. (2000). A Formal Ontology of Properties. In *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management* (pp. 97–112). Springer-Verlag.

Guarino, N., & Welty, C. A. (2009). An Overview of OntoClean. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 201–220). Springer Berlin Heidelberg.

Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. Netherlands: CTIT.

ISO 8373:2012. Robots and robotic devices - Vocabulary. (2012). ISO/TC 184/SC 2.

Lopez, M., Perez, A., & Juristo, N. (1997). METHONTOLOGY: from Ontological Art towards Ontological Engineering (pp. 33–40). Presented at the Proceedings of the AAAI97 Spring Symposium.

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001* (pp. 2–9). New York, NY, USA: ACM. doi:10.1145/505168.505170

Prestes, E., Carbonera, J. L., Fiorini, S. R., Jorge, V. A. M., Abel, M., Madhavan, R., … Schlenoff, C. (2013). Towards a core ontology for robotics and automation. *Robotics and Autonomous Systems*. doi:10.1016/j.robot.2013.04.005

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, *25*(1-2), 161–197.

An Incremental and Iterative Process for Ontology Building

Andre Menolli[1,3], H. Sofia Pinto[2], Sheila Reinehr[1], Andreia Malucelli[1]

[1]Programa de Pós-Graduação em Informática (PPGIa), Polytechnic School, Pontifícia Universidade Católica do Paraná – PUCPR, Curitiba, Brazil

[2]Instituto Superior Técnico (INESC-ID), Lisboa, Portugal

[3]Computer Science Department, State University of North Paraná, Bandeirantes, Brazil

menolli@uenp.edu.br, sofia@ontol.inesc-id.pt, sheila.reinehr@pucpr.br, malu@ppgia.pucpr.br,

Abstract. *The ontology development area has received some attention over the years. Methodologies focusing in diverse aspects of ontology development have emerged. Some of these methodologies are consolidated, presenting phases and activities. However, existing methodologies do not fully consider the ontology integration process. Therefore, based on METHONTOLOGY and a methodology for integrating ontologies we proposed an incremental and iterative process. We have used this process to develop an ontology following three iterations, which we present in this paper. Furthermore, we discuss the main features of the proposed process.*

1. Introduction

Knowledge representation through ontologies aims at capturing static domain knowledge in a generic way and provide a common agreement upon understanding of that domain, which may be reused and shared across applications and groups [Chandrasekar *et al*. 1999].

Ontologies can describe a hierarchy of concepts connected by subsumption relationships, a concept more aligned with taxonomies; or a structure where axioms are added to express relationships among concepts and to limit their intentional interpretations [Guarino 1998]. Axioms and subsumptions relationships allow the use of inference mechanism. Therefore, an ontology is a complex knowledge representation object, whose development requires the use of some methodology.

In this context, there are several and diverse methodologies focusing in various aspects of ontology development. The most representative ontology building methodologies are by [Uschold 1996], [Uschold and Grüninger 1996] and [Fernández et al. 1997]. Nevertheless, these methodologies present some limitations, as for instance they do not address ontology integration [Pinto 2000]. Therefore, specific methodologies for ontology integration were proposed, as [Gangemi et al. 1998] and [Pinto and Martins 2001]. Nevertheless, these methodologies focus on ontology integration, and despite of them enable work with other methodologies for development ontology, they do not detail how. Furthermore, all work mentioned above are methodologies, thus are more comprehensive than a process.

With the growing number of existing knowledge representation sources, a process to build new ontologies taking full advantage of existing sources is needed. Thus, in this paper we propose an iterative and incremental process for ontology development. This process considers the acquisition and use of external sources to develop each increment, and is concerned with the integration of ontologies developed in each increment.

The proposed process is based on METHONTOLOGY [Fernández et al. 1997] and in the methodology for integrating ontologies proposed by Pinto and Martins (2001), which describes a process of ontologies integration.

2. The Incremental and Iterative Process

The iterative process reduces the complexity of ontology development, since it divides it into small parts, and the incremental life cycle solves some problems, allowing the partial specification of requirements and makes the ontology grow by layers, allowing the inclusion of new definitions only when a new version is planned. Figure 1 shows the life cycle process, and each phase is described following.

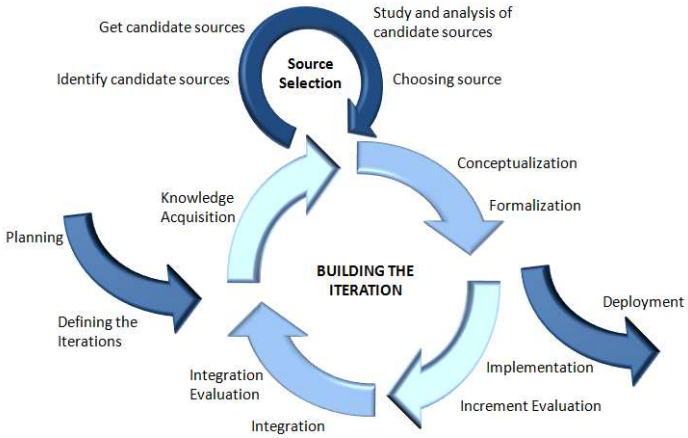


Figure 1. The Incremental and Iterative Life Cycle for Ontology Building

Planning

The planning phase is the first phase of ontology development. In this phase, the planning of whole ontology is done and the main goals are [Fernández et al. 1997]: (i) define the purpose of the ontology, including its intended uses, scenarios of use, end-users; (ii) define the level of formality of the implemented ontology, depending on the formality that will be used to codify the terms and their meaning; and (iii) define the ontology scope.

Defining the Iterations

The ontology, usually, is composed of several parts, which are aggregated to form the whole. So, it is important to define how many iterations will be needed to build the ontology, and the purpose of each one. This phase is extremely important, since the iterations defined here will guide the ontology development process.

Knowledge Acquisition

This phase was first defined by Fernández et al. (1997). In this phase, all knowledge about the domain must be acquired. However, instead of acquiring all knowledge to the whole ontology, we propose to divide and perform this phase for each increment. Thus in our process the knowledge acquisition is made incrementally, which facilitates the understanding of the subject.

Source Selection

Source Selection aims to select external sources that can be reused as base to develop the current increment. In this incremental and iterative process, each increment can be based in ontologies or other kinds of documents. This phase is composed of diverse activities, described following.

- *Identify candidate source:* the candidate sources should not be just ontologies, but any kind of knowledge representation. Among the main kind of knowledge representation, we suggest to use: catalog/id, terms/glossary, thesauri, frames, ontologies, and metadata specifications. This activity is subdivided into: *(1) finding available sources*, and *(2) choosing from the available sources which ones are possible candidates to be used*. To find possible sources, it is recommended to search in different locations, like ontology libraries and repositories of

standards organizations. To choose candidate sources one analyzes all available sources according to a series of features [Pinto and Martins 2001].

- *Get candidate source:* getting candidate sources includes not only their representations, but also, all available documentation. In some cases, this representation can be found in the literature (technical reports, books, thesis, etc.), or at least parts of it [Pinto and Martins 2001]. However, in most cases, only the implementation level representation of a source is available. Therefore, the reengineering process may be applied using the particular technique, according to the source chosen.

- *Study and analysis of candidate sources:* at this phase, it is important to study and analyze the sources to choose the best one. So, some criteria need be used according to Pinto and Martins (2000): (1) what knowledge is missing (concepts, relations, etc); (2) what knowledge should be removed; (3) which knowledge should be relocated; (4) which knowledge sources changes should be performed; (5) which documentation changes should be performed; (6) which terminology changes should be performed; (7) which definition changes should be made; and (8) which practices changes should be made.

- *Choosing source:* at this stage, and given the study and analysis of candidate sources performed by domain experts and ontologists, the final choices must be made. The source to be chosen and reused may lack knowledge, may require that some knowledge is removed, etc., that is, it may not exactly be what is needed. The best candidate source is the one that can better (more closely) or more easily (using less operations) be adapted to become the needed ontology [Pinto and Martins 2001].

Conceptualization

In this phase, the knowledge acquired is organized and structured using an independent knowledge representation. It is recommended that the knowledge domain is structured in a conceptual model that describes the problems and solutions in terms of the identified domain vocabulary [Fernández et al. 1997]. If an external source was selected as initial point to build the iteration, two additional activities are needed: adaptation and preparation to integration.

Adaptation focus on adapt the data from the external source to new domain. Many times an external source provides diverse concepts and attributes that are not need to the ontology that will be built. Preparing to integration it is needed to identify the assumptions and ontological commitments [Gruber 1995] that each increment should comply to.

Formalization

Transforms the conceptual model into a formal or semi-computable model, defining formal axioms. These axioms are introduced to constrain their interpretation and well-formed use [Pretorius 2004].

Implementation

In this phase, the increment is codified in a formal language such as OWL (Web Ontology Language).

Increment Evaluation

After implement the increment, the result ontology of the increment should be evaluated and analyzed. Furthermore, having an adequate design [Gruber 1995] and compliance with evaluation criteria [Gomez-Perez *et al*. 1995] the ontology should have a regular level of detail all over.

Integration

After the first iteration, the resulting ontology of the increment must be integrated with the ontologies created by the previous iterations. For that, one needs integration operations and integration oriented design criteria. Integration operations specify how knowledge from an

integrated ontology is going to be included and combined with knowledge in the resulting ontology, or modified before its inclusion [Pinto and Martins 2001].

Integration Evaluation

If it is not the first iteration, the integrated ontology should be evaluated and analyzed. None of the parts should have less level of detail than the required one or else the ontology would be useless, since it would not have sufficient knowledge represented. The resulting ontology should be consistent and coherent all over (although composed of knowledge from different ontologies).

3. Using the Process

This process was used to create the Unit of Organizational Learning Ontology (UOLO), and bellow are described the execution of all process stage to create the ontology.

Planning: This ontology aims at helping organize the content created in the company, specifically software development companies in units of organizational learning. It is based on educational units of learning, however considers organizational features.

Defining the Iterations: The UOLO was developed into three iterations: (1) organizational learning objects; (2) learning design; and (3) content package. The development of each increment was done following the activities outlined in Figure 1.

The first iteration generated the Ontology for Organizational Learning Object (OOLO) [Menolli *et al.* 2012].

- *Knowledge Representation*: in this phase the main Learning Objects Metadata were studied. From this study, the Learning Object Metadata (LOM) [IEEE 2002] was chosen as the base source to start developing the ontology proposed in this iteration, because it is a standard that facilitates search, acquisition, evaluation and use of LOs [Menolli et al. 2012].
- *Source Selection:* LOM Ontologies, and the complete documentation of LOM [IEEE. 2002] were gotten. Furthermore, FOAF (Friend of Friend) ontology also was gotten in this phase.
- *Conceptualization:* In this phase, all concepts and their properties were defined. This definition was done according to LOM standard, adapting it to our need and considering organizational features.
- *Formalization:* It was created a formal model that facilitates visualizing the taxonomy, covering axioms and properties.
- *Implementation:* The increment was implemented using the Protégé ontology editor and it was represented in OWL.

The other two iterations followed all the phases described in Figure 1. The second iteration implemented a learning design to help organizing materials previously produced in a manner that can enhance their understanding. So, it was based on IMS LD specification, that is a meta-language that describes all the elements of the design of a teaching-learning process, and drawn up by the IMS/LDWG work group [IMS 2003]. After implement this increment, it was integrated with the ontology created in the first iteration.

The third iteration created an ontology for Content Package concept. Content package describes the physical structure of the course defined by learning design. To define the content package concepts the IMS Content Packaging Specification [IMS 2004] was used. This increment was integrated with the ontologies produced in the first two iterations. The complete UOLO was generated as shown by Figure 2. Figure 2 (A) indicates the first increment, Figure 2 (B) the second increment and the Figure 2 (C) the third increment.

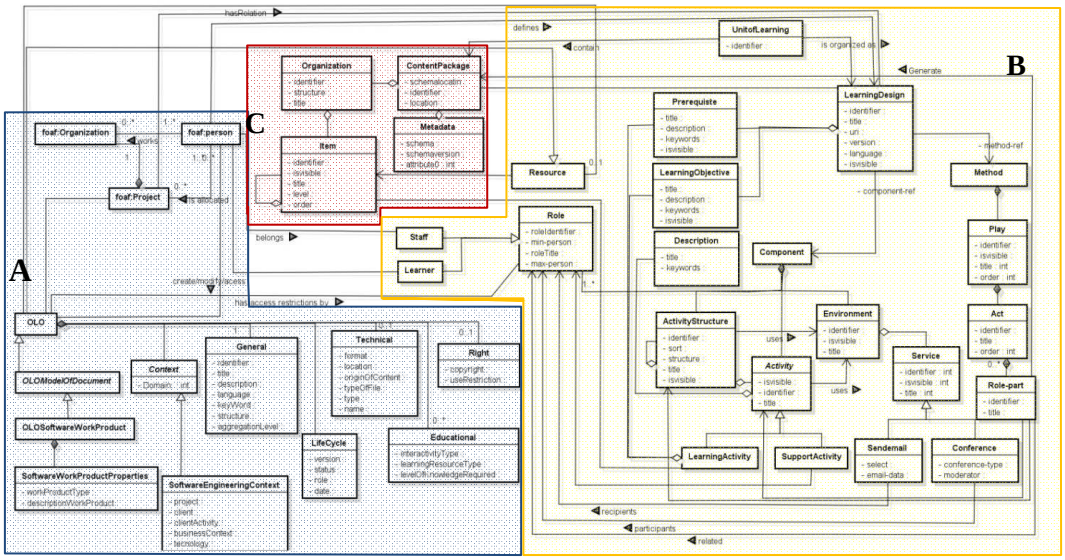


Figure 2. UOLO Concepts, Taxonomy and Relations

4. Discussion

In general, the phases that compose the life cycle tend to be performed following the order by which they were presented. If this order is performed, using the proposed life cycle the effort is divided between the phases.

The knowledge acquisition together with selection phase can require more effort than other approaches, since it is needed find and study several kinds of materials that can be used as base for the ontology; however, this effort should help to reduce the effort in the next phases. Using external sources to help modeling a concept model can reduce the effort of the conceptualization phase. Furthermore, finding external candidate sources, getting them, their evaluation and assessment for reuse purposes, and the choice of the most adequate one remain essential activities to be performed. This helps to create a more concise and consolidated model, since it is based on consensus knowledge.

The integration starts in the knowledge acquisition phase, and it continues for all other phases. Therefore, the integration is planned during all the increment, and if it is well performed, in integration phase, the ontology is just implemented together with the ontology created previously, and in the next phase, the integrated ontology is evaluated. Each increment must be evaluated individually, and after that it must be integrated with the ontology, and at the end evaluate the resulting ontology.

This process facilitates to find external sources to be reused. Moreover, the ontologist is forced to focus on the most critical issues, reducing risks during development; furthermore, the iterative and incremental development enables a continuous assessment of the project status. Finally, develop each increment is simpler than develop the whole ontology. As main limitation, the domain must be known and the scope limited, facilitating the iterations identifications.

5. Final Considerations

In this paper we describe an incremental and iterative process to ontology building. Furthermore, we describe the process life cycle and its phases. An incremental ontology was created using the proposed process, and as main advantages we identified the ease of use external sources, focusing on the most critical issues and the continuous and objective assessment of the project status. However, this process should be used only when the ontologist knows the domain, and he/she is sure that the ontology has more than one iteration.

The proposed process instantiate a particular integration process, using the phases and activities proposed by other ontology methodologies. The process reuses external material to build

each increment. For this, we used and adapted the activities defined by [Pinto and Martins 2001], that help to evaluate and choose the best sources from the identified sources. Furthermore, it integrates the activities to reuse sources with the phases proposed in the METHONTOLOGY. The process puts special emphasis to the quality of the final ontology, since we propose to evaluate each increment as well as the whole ontology.

Acknowledgments

The authors thank the Fundação Araucária of Paraná, Brazil and the Portuguese Fundação para a Ciência e Tecnologia through the financial support of INESC-ID funding under project PEst-OE/EEI/LA0021/2013.

References

Chandrasekaran, B., Josepheson, J. and Benjamins. V.R. (1999) “Ontologies: What are they? Why do we need them?”, In *IEEE Intelligent Systems*, v. 14, n. 1, pages 20-26.

Fernández, M., Gómez-Pérez, M. and Juristo, N. (1997) “Methontology: From Ontological art Towards Ontological Engineering”. Ontological Engineering, Stanford, Califórnia.

Gangemi, A., Pisanelli D, and Steve. G. (1998) “Ontology Integration: Experiences with Medical Terminologies. In N. Guarino (ed.), Formal Ontology in Information Systems, pages 163-178.

Gómez-Pérez, M., Juristo, N. and Pazos J. (1995) “ Evaluation and Assessment of the Knowledge Sharing Technology”, In *N. Mars (ed.), Towards Very Large Knowledge Bases*, pages 289-296. IOS Press.

Gruber. T. (1995) “Towards Principles for the Design of Ontologies for Knowledge Sharing”, In *International Journal of Human Computer Studies*, v. 43, n. 5-6, pages, 907-928.

Guarino, N. (1998) “Formal Ontology and Information Systems”, In *Proceedings of Formal Ontology in Information System*, pages. 3-15.

IEEE (2002) “Draft Standard for Learning Object Metadata”, Available on http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf July, 2002.

IMS (2003) “IMS Learning Design Information Model”, Version 1.0 Final Specification. Available on: http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

IMS (2004) “IMS Content Packaging Information Model”, Version 1.1.4, IMS Global Learning. Available on: http://www.imsglobal.org/content/packaging/cpv1p1p2/imscp_infov1p1p2.html.

Menolli, A. L., Reinehr., S. and Malucell, A. (2012) “Ontology for Organizational Learning Objects based on LOM Standard”, In *Proceedings of the Latin American Conference en Informática* (Colombia, 2012).

Pinto H. S. and Martins, J. P. (2000) “Reusing Ontologies”, In *Proc. of AAAI2000 Spring Symposium Series, Workshop on Bringing Knowledge to Business Processes*, pages 77-84.

Pinto H. S. and Martins, J. P. (2001) “A methodology for ontology integration”, In Proceedings of the 1st international conference on Knowledge capture, pages 131-138, Canada.

Pretorius, A. J. (2004) “Ontologies - Introduction and Overview”, In *Mosaic A Journal For The Interdisciplinary Study Of Literature*, pages. 1-13.

Uschold, M. (1996) “Building Ontologies: Towards A Unified Methodology”, In *Expert Systems*, v. 96.

Uschold, M. and Grüninger, M. (1996) “Ontologies: Principles Methods and Applications”, In *Knowledge Sharing and Review*, V. 2.

OntoAlign++: a Combined Strategy for Improving Ontologies Alignment

Miguel Gabriel Prazeres Carvalho[1], Maria Luiza Machado Campos[1], Linair Maria Campos[2], Maria Cláudia Cavalcanti[3]

Programa de Pós Graduação em Informática– UFRJ[1], Rio de Janeiro,RJ,Brasil
Programa de Pós Graduação em Ciência da Informação – UFF[2], Niterói, RJ, Brasil
Programa de Pós Graduação em Sistemas e Computação-IME[3], Rio de Janeiro,RJ,Brasil

{miguelgabriel[1],mluiza[1]}@ufrj.br, linair@cisi.coppe.ufrj.br[2], yoko@ime.eb.br[3]

Abstract. *Ontology reuse is very important nowadays but, more specifically, ontology alignment still represents a challenge, despite the proposal of a great number of techniques and tools that implement it. This paper presents an approach that builds upon two already existent techniques. It considers both the enrichment of the ontologies with implicit terms and relationships contained on the ontologies terms definitions and on associating concepts of the ontologies to categories of foundational ontologies. Besides confirming the improvement on alignment results when using each of these approaches, our experiments showed even better results when these techniques were applied together.*

1. Introduction

In recent years, the use of ontologies has greatly increased in different areas, from serving as a basis for conceptual modeling, formally defining an abstraction of a given perspective of reality, to supporting resource interoperability and knowledge discovery from multiple sources.

However, due to an increasing demand, many ontologies are built in an ad hoc manner, lacking a systematic approach for their development. This contributes to several problems when using those ontologies, mainly compatibility and interoperability between them (Kohler et al. 2006). Also, inconsistencies in ontologies structure can lead to errors in the alignment process, mistakenly associating non similar terms (Silva et al. 2011; Kohler et al. 2006; Smith, Kohler, Kumar 2004). Several studies attempt to address these problems (Ehrig 2007; Lambrix and Tan 2006; Kalfoglou and Schorlemmer, 2003). More recently, our research group has conducted two studies in this area (Silva et al. 2011; Carvalho et al. 2011) considering strategies for complementing the ontologies explicit knowledge, by applying some previous treatment on selected ontologies before the alignment process, providing in both cases a significant improvement in the results. This paper aims to merge these two approaches

creating a third one that is analyzed to collect evidences that it is possible to further improve the alignment process.

The remaining of this paper is organized as follows. Section 2 presents an overview of ontologies alignment strategies. Section 3 gives a brief summary of the approaches of Silva et al. (2011) and Carvalho et al. (2011). Section 4 presents the experimental analysis conducted on the biomedical ontologies scenario and discusses the results obtained. Finally, Section 5 presents concluding remarks and future work.

2. Ontology Alignment Techniques

In the context of ontologies reuse, the alignment process constitutes an important instrument for the combination of the information contained in multiple but related ontologies, identifying similarities between their individual elements. It is considered the process of establishing one-to-one equality relations between the terms of two ontologies from the same domain (Ehrig 2007).

There are many available alignment tools that implement a combination of alignment techniques proposed on different approaches throughout the years. These tools consider similarity as a measure associated to elements from the ontologies being aligned, that corresponds to a numeric value indicating how similar or different the elements are. Most of the tools calculate similarity based on a combination of alignment techniques (Euzenat and Shvaiko 2007). For this paper, we have focused on techniques that complement the existing terms and structure with concepts and relations already available in definitions or other ontology elements, as well as techniques that use top-level ontologies (Guizzardi 2009) to express the ontological commitment of the ontology conceptualization.

3. ONTOALIGN++ and approaches from Silva et al. and Carvalho et al.

In Silva et al. (2011), before the alignment itself, a preparation step associates terms from the top three levels of the domain ontology to terms from the foundational ontology used – BFO (2012). This association helps to prevent incorrect similarity assumptions in the alignment process, restricting the indication of equivalent terms to those derived from the same meta-category, i.e. those having the same conceptual nature. As an additional customization, it also takes into account previous alignments, which serve as a reference to validate correct alignments, and also to discard incorrect ones, avoiding that these are repeatedly presented to user validation afterwards. After

this, other preliminary steps are also contemplated, such as fragment extraction and cleaning. In the ontology alignment step, after source and target ontologies are prepared, the alignment is then applied, based on the NOM (Naive Ontology Mapping) approach used by the FOAM tool (Ehrig and Sure 2005), but customized with selected measures, foundational ontologies and previous alignments.

The work of Carvalho et al. (2011) explores implicit information contained in ontologies (especially those contained in the definition field) and how this information can be extracted aiming at the improvement of various processes, including the alignment. This approach uses data mining techniques in order to extract new terms and relationships in ontologies, to allow for their semantic improvement, by complementing the ontologies with these elements. It uses linguistic tools, as GATE (Bontcheva et al. 2003) and NLTK (Bird et al. 2009), and is implemented through the EI-ONTO tool, which provides support for all the steps of the approach. The approach includes two macro-steps. The first macro-step has the goal of studying the corpus and is divided into three steps: (i) transform the corpus; (ii) treat the corpus; and (iii) categorize the corpus. The second macro-step is to find corpus patterns. It uses a machine learning strategy and aims at finding patterns in the definition and in the nomenclature of terms. After these steps, the extracted terms and relationships are temporarily added to the ontology, enriching the domain knowledge already represented, and improving the alignment results, as shown in Carvalho et al. (2011).

The ONTOALIGN++ approach takes advantage of the increased expressiveness derived from both approaches described previously. First, an existing ontology can be enriched by complementing it with further terms and relationships that are "implicitly" represented in the terms definitions. Secondly, applying Silva's approach, a complementary semantic layer can be added to this enriched ontology, extending the ontology with a more precise representation of existing concepts. Using concepts from the foundation ontology, terms from the enriched ontology have their ontological commitment made explicit.

4. Experimentation and Results Analysis

Aiming to explore the chosen strategies and verify that their combined use enables real gain in the alignment process, we conducted an empirical study. Moreover, we added as a secondary objective of this study the verification of the efficiency of the individual use

of each of these approaches when considering an improved version of the original ontologies (they had been enhanced by OBO since the last experiments conducted by the authors). The goal of this verification is to check if the use of Silva´s et al. (2011) and Carvalho´s et al. (2011) approaches still provide an effective improvement in the alignment process, even with new improved versions of original OBO ontologies used.

Adopting an empirical approach, we have verified the efficiency of the approaches both used individually and combined. For this evaluation we have chosen two ontologies to be aligned, the Biological Process Ontology (BPO) and INOH Event Ontology. We executed four experiments and producing four corresponding results: (i) Ontologies aligned without any additional information; (ii) Ontologies aligned using Carvalho´s et al. approach (2011); (iii) Ontologies aligned using Silva´s et al. approach (2010); (iv) Ontologies aligned using Ontoalign++ approach. The first experiment was carried out without additional information. For the second experiment, we have applied Carvalho´s approach (2011) on each ontology, identifying 198 relationships in the BPO and 59 relationships in the INOH. These relationships were manually validated, resulting on 187 BPO relationships selected as valid against 54 relationships in INOH. For the third experiment we applied the approach of Silva et al. (2011) to the original ontologies, using the strategy described in that work. The foundational ontology chosen was again the BFO, for its adequacy to the biomedical area. Terms from the first three levels (as defined in the approach) of the original ontologies where linked to BFO terms, resulting on two extended ontologies. For the last experiment we have combined both the enrichment and extension to the original ontologies. We first applied the approach of Carvalho et al. (2011), in fact, using the same enriched ontologies of the second experiment. After that, we associated these enriched ontologies to the terms of BFO, as in Silva et al. (2011).

Having prepared the ontologies for each experiment, we used the FOAM tool for executing the alignments, with the following parameters: alignment–fully Automatic; number of iterations - 10; cutoff value: - 0.97; strategy – Decision Tree (Decision Tree). After the alignments, the resulting matches were independently validated by two biologists with expertise in the area of genome sequencing.

4.1 Results analysis

The results were tabulated and are described in Table 1, where the alignments numbers correspond to the experiments as described previously. From these results, there are evidences that the combined use of the two approaches enhances the alignment process. In addition to the increase of pairs correctly aligned, there is also a decrease in the occurrence of pairs aligned with weak relations. Another important result is the improvement on alignments 2 and 3 when compared to alignment 1, confirming that Carvalho's et al. (2011) and Silva's et al. (2011) approaches, even when used individually, are important in order to increase the number of correctly aligned pairs. In this scenario, our evidences suggest that both approaches contribute to enhance the precision of the ontologies alignment process, and, more importantly, they can be combined to reach even better results. In fact, they are complementary to each other.

Table 1: Alignment Results

Classification	Results			
Degree	Alignment 1	Alignment 2	Alignment 3	Alignment 4
5 – correct	37	45	43	49
4 – strong relation	5	7	8	8
3 – medium relation	5	5	5	5
2 – weak relation	7	1	3	1
1 – incorrect	1	1	1	1
Total	55	59	60	64

Observing the alignments, we noticed that some of the errors derived from imprecisions on the original ontologies, as some *is_a* relationships were mistakenly represented as *part-of* relationships, and vice-versa. Also, there are gaps in the specialization hierarchies, which induce errors in the alignment process. In this last case, we have evidences that some of these gaps could be removed by refining our enrichment strategy so that more intermediary *is_a* relationships could be extracted from the definitions.

5. Conclusion

Even with some quality improvement incorporated more recently on existing ontologies, their reuse still present considerable challenges. Most often, when trying to reconcile overlapping domain ontologies it is not trivial to solve ambiguities and to identify similarities as main commitments that underline an ontology conceptualization which has not been properly externalized. Besides reevaluating two successful approaches used to improve the alignment of ontologies, this work also aimed at

showing evidences of the advantage of combining them. The executed experiments yielded not only an increase in the number of pairs aligned as well as a decrease in the number of false alignments. As future work, other possibilities could be explored, such as exploring associating terms from other levels of the domain ontology as well as exploring other extraction strategies and trying semi-automatic mechanisms for associating to the top-level ontology.

References

BFO (2012) Basic Formal Ontology. Available at: http://www.ifomis.org/bfo.

Bird, S., Klein, E., Loper, E. (2009) Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit. Sebastopol, CA: O'Reilly.

Bontcheva, K., Kiryakov, A., Cunninghan, H., Popov, B., Dimitrov, M. (2003) Semantic web enabled, open source language technology. In EACL workshop on Language Technology and the Semantic Web: NLP and XML, Hungary.

Carvalho, M.G.P., Campos, L.M., Braganholo, V.P., Campos, M.L.M., Campos, M.L.A. (2011) Extracting New Relations to Improve Ontology Reuse. Journal of Information and Data Management, v. 2, p. 541-556.

Ehrig, M. (2007) Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond), Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Ehrig, M. and Sure, Y. (2005) FOAM - Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative. In: Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Canada.

Euzenat, J. and Shvaiko, P. (2007) Ontology matching, Springer Verlag, Berlin, Germany.

Guizzardi, G. (2009) Ontology-Driven Conceptual Modeling - II Seminario de Pesquisa em Ontologia no Brasil. Available at ontobra.comp.ime.eb.br/apresentacoes/curso2.

Kalfoglou, Y. and Schorlemmer, M. (2003) Ontology mapping: the state of the art. Knowledge Engineering Review, v.18, n.1, p.1-31.

Köhler, J., Munn, K., Ruegg, A., Skusa, A., Smith, B. (2006) Quality Control for Terms and Definitions in Ontologies and Taxonomies. BMC Bioinformatics, v.7, n.212, p.1-12.

Lambrix, P. and TAN, H. (2006) SAMBO - A System for Aligning and Merging Biomedical Ontologies .Web Semantics: Science, Services and Agents on the World Wide Web, v.4, n.3, p.196-206.

Silva, V.S., Campos, M.L.M., Silva, J.C.P., Cavalcanti, M.C. (2011) An Approach for the Alignment of Biomedical Ontologies based on Foundational Ontologies. Journal of Information and Data Management, v. 2, p. 557-572.

Smith, B.; Köhler, J.; Kumar, A. (2004) On the Application of Formal Principles to Life Science Data: A Case Study in the Gene Ontology. In Database Integration in the Life Sciences, p.1-17.